

技術分野

【0001】

- 5 この発明は、画像データに所定の画像処理を施して画像を出力する技術に関し、詳しくは、画像処理が施された画像データを画像出力装置に迅速に転送することによって、画像の出力を迅速化する技術に関する。

背景技術

10 【0002】

- 印刷媒体や液晶画面といった各種の出力媒体上にドットを形成することで画像を表現する画像出力装置は、各種画像機器の出力装置として広く使用されている。これら画像出力装置では、画像は画素と呼ばれる小さな領域に細分された状態で扱われており、ドットはこれら画素に形成される。ドットを画素に形成した場合、  
15 もちろん個々の画素については、ドットが形成されているか、いないかのいずれかの状態しか取り得ない。しかし、画像全体として見れば、ドットが密に形成されている領域や、まばらに形成されている領域を生じさせることが可能であり、従って、ドットの形成密度を変えることによって画像を出力することが可能となる。例えば、印刷用紙上に黒いインクのドットを形成する場合、ドットが密に形  
20 成されている領域は暗く見えるし、逆にドットがまばらに形成されている領域は明るく見える。また、液晶画面に輝点のドットを形成する場合、ドットが密に形成された領域は明るく見え、まばらに形成された領域は暗く見える。従って、ドットの形成密度を適切に制御してやれば、多階調の画像を表示させることが可能となる。

25 【0003】

- ドットの形成密度を制御するためのデータは、出力しようとしている画像のデータに対して、所定の画像処理を施すことによって発生させる。得られたデータは画像処理装置に供給され、画像処理装置では、供給されたデータに従って各画

素にドットを形成する。この結果、出力媒体上に適切な密度でドットが形成され、画像が出力されることになる。

#### 【0004】

近年では、これら画像出力装置には、出力画像の高画質化や大画像化が要請されるようになってきた。高画質化の要請に対しては、画像をより細かな画素に分割することが効果的である。こうして画素を小さくすれば、画素に形成されるドットが目立たなくなって画質を向上させることができる（例えば、特開2000-115716号公報）。また、大画像化の要請に対しては、画素数の増加によって対応することができる。もちろん、個々の画素を大きくすることによっても出力画像を大きくすることはできるが、これでは画質の低下を招いてしまうので、大型化の要請に対しては画素数を増加させることが効果的である。

#### 【0005】

しかし、出力画像の高画質化や大画像化の要請に応えようとする、いきおい画像を迅速に出力することが困難になるという問題があった。すなわち、大画像化のために画素数を増加させた場合はもちろんのこと、画像の高画質化のために個々の画素を小さくした場合も、結果として1つの画像に含まれる画素数が増加する。前述したように、画像出力装置は、ドットの形成を制御するデータの供給を受けて画像を出力するので、1つの画像に含まれる画素数が増加すると、データの供給に要する時間が増加してしまい、画像を迅速に出力することが困難となるのである。

#### サマリー

#### 【0006】

この発明は、従来技術における上述した課題を踏まえ、ドットの形成を制御するためのデータを画像出力装置に迅速に供給することで、画像の出力を迅速化させることを目的とする。

#### 【0007】

上述した目的を達成するために、本発明の画像出力制御システムは、次の構成

を採用した。すなわち、

画像データに所定の画像処理を施す画像処理装置と、単位面積当たりの濃度が異なる複数種類のドットを該画像処理の結果に基づいて形成することにより画像を出力する画像出力装置と、を備える画像出力制御システムであって、

5 前記画像処理装置は、

前記画像を構成する画素が複数個ずつまとめられた画素群について、該画素群内に形成されるドットの個数を前記画像データに基づいて前記ドットの種類毎に決定するドット個数決定手段と、

前記各種ドットについて決定したドット個数のデータを、前記画像出力装置  
10 に前記画素群毎に出力する個数データ出力手段と

を備えており、

前記画像出力装置は、

前記各種ドットについてのドット個数のデータを前記各画素群毎に受け取る  
個数データ受取手段と、

15 前記画素群内で各画素にドットが形成される画素の序列を前記画素群毎に特定する序列特定手段と、

前記各種ドットについてのドット個数のデータと、前記特定した序列とに基づいて、前記画素群内で該各種ドットが形成される画素位置を該画素群毎に決定する画素位置決定手段と、

20 前記決定した画素位置に基づいて前記各種のドットを形成するドット形成手段と

を備えることを要旨とする。

【0008】

また、上記の画像出力制御システムに対応する本発明の画像出力制御方法は、  
25 画像データに所定の画像処理を施した後、単位面積当たりの濃度の異なる複数種類のドットを該画像処理の結果に基づいて形成することにより画像を出力する画像出力制御方法であって、

前記画像を構成する複数の画素が所定の複数個ずつまとめられた画素群につい

て、該画素群内に形成されるドットの個数を前記画像データに基づいて前記ドットの種類毎に決定する第1の工程と、

前記画素群内で各画素にドットが形成される画素の序列を前記画素群毎に特定する第2の工程と、

- 5 前記決定した各種ドットの個数と前記特定した序列とに基づいて、前記画素群内で各種ドットが形成される画素位置を該画素群毎に決定する第3の工程と、

前記決定した画素位置に基づいて前記出力媒体上に前記各種のドットを形成する第4の工程と

を備えることを要旨とする。

10 【0009】

- かかる本発明の画像出力制御システムおよび画像出力制御方法においては、複数の画素から構成される画素群毎に、該画素群内に形成すべき各種ドットの個数のデータを生成する。各種ドットを形成するに際しては、画素群毎に序列を特定し、該特定した序列と、画素群について受け取った各種ドットの個数のデータと
- 15 に基づいて、該画素群内で各種ドットが形成される画素位置を決定する。こうして決定した画素位置に基づいて各種のドットを形成する。

【0010】

- 画素群内に形成するドットの個数は、画素群内の各画素についてのドット形成の有無よりも僅かなデータ量で表すことができる。従って、画像データを、画素
- 20 群のドット個数のデータに一旦変換しておけば、出力するデータ量が減少するので速やかに出力することができる。このため、例えば画素数の多い画像であっても、僅かな時間でデータの出力が完了して、迅速に画像を出力することが可能となる。特に、複数種類のドットを形成可能な場合、ドットの種類毎にドット形成の有無を出力したのではデータ量が多くなってしまう。従って、各種ドットの個数のデ
- 25 ータに変換してから出力してやれば、出力するデータ量を大きく減少させることができ、延いては、画像の出力に要する時間を大幅に短縮することが可能となる。

【0011】

また、画素群内での画素位置は、各種ドットの個数のデータと、画素の序列、

すなわち、該画素群内で各画素にどのような順序でドットが形成されるかを表す  
5 序列とに基づいて決定される。ここで、画素位置の決定に際しては、複数記憶さ  
れている序列の中から、画素群毎に選択された序列を用いて画素位置が決定され  
る。このため、例え、各種ドットが同じ個数ずつ形成される画素群が連続した場  
10 合でも、画素群毎に異なる画素位置にドットが形成されることになる。従って、  
規則的なパターンで形成されているドットが目立ってしまい、画質を悪化させる  
おそれを回避することができる。

#### 【0012】

こうした画像出力制御システムにおいては、各種ドットについてのドット個数  
10 のデータをそのまま出力するのではなく、次のように符号に変換してから出力す  
ることとしても良い。先ず、各種ドットについてのドット個数の組み合わせ毎に、  
所定の符号を対応付けたドット個数組対応表を記憶しておく。そして、ある画素  
群について各種ドットの個数が決定されたら、その個数の組合せを、該ドット個  
数組対応表に従って対応する符号に変換し、得られた符号を画像出力装置に出力  
15 する。画像出力装置には、符号毎に、各種ドットの個数の組合せを対応付けた符  
号対応表を、予め記憶しておく。そして、出力された符号を受け取ると、該符号  
対応表を参照することによって、各種ドットの個数に変換することとしても良い。

#### 【0013】

画素群内に形成するドットの個数をドットの種類毎に出力するよりも、符号に  
20 変換してから出力した方が、データ量が小さくなるので、データを迅速に出力す  
ることができる。もちろん、ドット個数のデータを符号に変換して出力する場合、  
各種ドットの個数の組合せを符号に変換する処理や、逆に符号を各種ドットの個  
数の組合せに変換する処理が必要となるが、こうした変換は、各種ドットの個数  
の組合せと符号とを対応付けた変換表を予め記憶しておくことで、迅速に実施す  
25 ることができる。従って、各種ドットの個数のデータを符号に変換してから出力  
してやれば、データの出力に要する時間を短縮することができ、延いては画像を  
迅速に出力することが可能となるので好ましい。

#### 【0014】

あるいは、こうした画像出力制御システムにおいては、画素群内で各種ドットが形成される画素位置を決定するに際して、単位面積当たりの濃度が高い種類のドットから順番に画素位置を決定することとしてもよい。

【0015】

- 5 単位面積当たりの濃度が高いドットは目立ち易いので、こうした高階調値のドットについてはドットが偏って形成されることの無いように、画素位置を決定してやる必要がある。ところが、異なる種類のドットについて画素位置が既に決定されていると、これらの画素を避けながら画素位置を決定しなければならないので、画素位置の決定が制約を受けることになって、高階調値のドットが偏ってしま
- 10 まい画質が低下するおそれがある。これに対して、単位面積当たりの濃度が高いドットから順番に画素位置を決定してやれば、目立ち易いドットほど、他のドットの制約を受けずに画素位置を決定することができるので、良好な画質を維持することが可能となる。

【0016】

- 15 また、上述した画像出力制御システムおよび画像出力制御方法においては、単位面積当たりの濃度の異なる複数種類のドットについて、画素群内に形成するドットの個数を決定するに際して、次のようにして決定しても良い。先ず、画素群を構成する画素数に相当する複数個の閾値からなる閾値の組を記憶しておく。次いで、画像データに基づいて画素群毎に、第1のドット密度データと第2のド
- 20 ット密度データとを生成する。ここで、第1のドット密度データとは、各種ドットの中で単位面積当たりの濃度の最も高い第1のドットが画素群内に形成される密度に対応するデータである。また、第2のドット密度データとは、各種ドットの中で単位面積当たりの濃度が2番目に高い第2のドットまたは前記第1のドットのいずれかのドットが、画素群内に形成される密度に対応するデータである。そ
- 25 して、該第1のドット密度データと該閾値組に含まれる閾値とを比較して、第1のドット密度データよりも小さな閾値の個数を、その画素群内に形成される第1のドットの個数と決定する。
- また、第2のドット密度データと前記閾値組に含まれる閾値とを比較して、該第

2のドット密度データよりも小さな閾値の個数と該第1のドットの個数とに基づいて、その画素群内に形成される第2のドットの個数を決定する。このとき、第2のドット密度データは、第1のドット密度データより大きな閾値とだけ比較して、該第2のドット密度データより小さな閾値の個数を検出する。

5           【0017】

こうすれば、第1のドット密度データおよび第2のドット密度データと閾値とを比較することでドットの個数を決定することが可能であり、画素群内に形成するドットの個数を簡便に決定することができる。また、第2のドット密度データは、第1のドットが形成される密度に対応する第1のドット密度データよりも大きな値となるので、第1のドット密度データよりも小さな閾値であれば、比較するまでもなく第2のドット密度データより小さいことは明らかである。従って、第1のドットの個数を決定した後第2のドットの個数を決定することとして、第2のドット密度データは、第1のドット密度データより大きな閾値とだけ比較することとすれば、ドットの個数を迅速に決定することが可能となるので好適である。

          【0018】

尚、複数種類の閾値の組を記憶しておき、画素群毎に1の閾値組を選択して、各種ドットの個数を決定することとしても良い。異なる閾値組を用いた場合、ドット密度データが同じであっても決定されるドット個数が異なる場合がある。従って、画素群毎に閾値組を選択しながらドットの個数を決定してやれば、得られたドットの個数には何がしかの変動が加わることになり、その結果、規則的なパターンでドットが形成されてしまうことを容易に回避できるという利点が得られる。

          【0019】

25   このようにしてドットの個数を決定する場合、複数個の閾値を前記閾値組の中で大きさの順番とともに記憶しておき、第2のドットの個数を決定する際には、第1のドット密度データよりも大きな閾値を該大きさの順番に基づいて選択することとしてもよい。例えば、順番が大きくなるほど閾値の値が大きくなるとして、

第1のドット密度データより小さな閾値がN個あった場合には、(N+1)番目以降の閾値と第2のドット密度データとを比較する。

#### 【0020】

こうすれば、第2のドット密度データと比較する閾値を簡便に選択することができ、延いては、ドットの個数を簡便に且つ迅速に決定することが可能となるので好ましい。

#### 【0021】

また、こうした画像出力制御システムにおいては、直前に決定した第1のドットの個数に基づいて閾値の番号を決定し、決定した番号の閾値から第1のドット密度データとの比較を開始することとしてもよい。例えば、直前にドット個数を決定した画素群では、第1のドット個数がM個である場合は、M番目、あるいは(M-1)番目、(M+1)番目などのように、Mを基準として定められる番号の閾値から比較を開始することとしてもよい。

#### 【0022】

画像データは、一般に、隣接する画素には近似したデータが割り当てられる傾向があるので、隣接する画素群についても近似した個数のドットが形成される傾向にある。特に、単位面積当たりの濃度の最も高い第1のドットについては、隣接する画素群間で同じドット個数となる傾向が強く、例えばある画素群で第1のドットをM個形成すると判断されていれば、続く画素群でもM個形成すると判断される可能性が高い。このことから、第1のドット密度データを閾値と比較するに際しては、直前に決定した第1のドットの個数に基づいて閾値の番号を決定し、決定した番号の閾値から比較を始めれば、第1のドット密度データより小さな閾値の個数を僅かな比較回数で決定することができる。その結果、画素群に形成するドットの個数を迅速に決定することが可能となるので好ましい。

#### 【0023】

更に、こうした画像出力制御システムにおいては、閾値組の中に、第1のドット密度データより小さな閾値が存在しない場合には、直前に決定した第2のドットの個数に基づいて閾値の番号を決定し、決定した番号の閾値から第2のドット



密度データとの比較を開始することとしてもよい。

【0024】

前述したように、隣接する画素群には近似する個数のドットが形成される傾向にあるから、ある画素群に形成されるドットの個数が、第1のドットが0個、第2のドットがM個であれば、続く画素群についても第1のドットは0個、第2のドットはM個となる可能性が高いと考えられる。従って、第1のドット密度データより小さな閾値が存在せず、第2のドット密度データと比較する場合には、直前に決定した第2のドットの個数に基づいて閾値の番号を決定し、決定した番号の閾値から比較を始めれば、僅かな比較回数で第2のドットの個数を決定することが可能となるので好ましい。

【0025】

また、上記の目的を達成するために、本発明の画像出力装置は次の構成を採用した。すなわち、

所定の画像処理が施された画像データを受け取って、単位面積当たりの濃度の異なる複数種類のドットを該画像データに基づいて形成することにより画像を出力する画像出力装置であって、

前記画像を構成する複数の画素を所定の複数個ずつまとめた画素群の各々について、該画素群内に形成すべき各種ドットの個数のデータを、前記画像データとして受け取る個数データ受取手段と、

前記画素群内で各画素にドットが形成される画素の序列を前記画素群毎に特定する序列特定手段と、

前記各種ドットについてのドット個数のデータと、前記特定した序列とに基づいて、前記画素群内で該各種ドットが形成される画素位置を該画素群毎に決定する画素位置決定手段と、

前記決定した画素位置に基づいて、前記出力媒体上に前記各種のドットを形成するドット形成手段と

を備えることを要旨とする。

【0026】

更に、上記の画像出力装置に対応する本発明の画像出力方法は、

単位面積当たりの濃度が異なる複数種類のドットを、所定の画像処理が施された画像データに基づいて形成することにより画像を出力する画像出力方法であって、

- 5 前記画像データとして、前記画像を構成する複数の画素が所定の複数個ずつ画素群としてまとめられた状態で、該画素群内に形成すべき各種ドットの個数のデータを受け取る工程（A）と、

前記画素群内で各画素にドットが形成される画素の序列を、前記個数のデータを受け取った画素群毎に特定する工程（B）と、

- 10 前記各種ドットについて受け取った個数のデータと前記特定した序列とに基づいて、前記画素群内で該各種ドットが形成される画素位置を、該画素群毎に決定する工程（C）と、

前記決定した画素位置に基づいて、前記出力媒体上に前記各種のドットを形成する工程（D）と

- 15 を備えることを要旨とする。

#### 【0027】

- こうした画像出力装置および画像出力方法においては、各種ドットについて画素群に形成すべきドット個数のデータを受け取る。そして、このデータに基づいて画素群内で各種ドットが形成される画素位置を決定し、決定した画素位置に各種のドットを形成することによって画像を出力する。
- 20

#### 【0028】

- 前述したように、画素群内に形成するドットの個数は、画素群内の各画素についてのドットの形成状況よりも僅かなデータ量で表すことができる。特に、複数種類のドットを形成可能な場合には、ドットの種類毎に形成状況を表すよりも、各種ドットの個数を表した方がデータ量は少なくなる。このため、上述したように、各種ドットの個数のデータを受け取って画像を出力することとすれば、例えば画素数の多い画像であっても、データを迅速に受け取ることができ、延いては画像を迅速に出力することが可能となる。
- 25

### 【0029】

- また、画素群内での画素位置を決定するために用いられる画素の序列は、画素群毎に特定されるので、例え、各種ドットが同じ個数ずつ形成される画素群が連続した場合でも、画素群毎に異なる画素位置が決定される。このため、規則的な
- 5 パターンで形成されたドットが目立って、画質を悪化させるおそれがない。

### 【0030】

- こうした画像出力装置においては、各種ドットについてのドット個数のデータをそのまま受け取るのではなく、次のようにして受け取ることとしても良い。先ず、各種ドットについてのドット個数の組み合わせが、所定の符号に対応付けられて
- 10 れている符号対応表を記憶しておく。そして、画素群に形成される各種ドットの個数の組合せが該符号に変換されたデータを、画像データとして受け取る。次いで、受け取った画像データを、該符号対応表を参照することによって、各種ドットの個数に変換することとしても良い。

### 【0031】

- 15 前述したように、画素群内に形成するドットの個数をドットの種類毎に受け取るよりも、符号に変換された状態で受け取る方がデータ量が小さくなるので、データを迅速に受け取ることができ、延いては画像を迅速に出力することが可能となるので好ましい。

### 【0032】

- 20 あるいは、こうした画像出力装置においては、画素群内にドットが形成される画素位置を、各種ドットについて決定するに際して、単位面積当たりの濃度の高い種類のドットから順番に画素位置を決定することとしてもよい。

### 【0033】

- 前述したように、単位面積当たりの濃度が高いドットは目立ち易いので、ドット
- 25 トが偏って形成されることの無いように画素位置を決定する必要がある。従って、単位面積当たりの濃度の高い種類のドットから順番に画素位置を決定してやれば、目立ち易いドットほど、他のドットの制約を受けずに画素位置を決定することができ、良好な画質を維持することが可能となって好ましい。

#### 【0034】

更に、上記の目的を達成するために、本発明の画像処理装置は、次の構成を採用した。すなわち、

単位面積当たりの濃度の異なる各種ドットを形成して画像を出力する画像出力  
5 装置に供給されて、該出力装置内で該各種ドットの形成を制御するために用いられる制御データを、該画像を表す画像データに所定の画像処理を施して生成する画像処理装置であって、

前記画像を構成する複数の画素が所定の複数個ずつまとめられた画素群について、該画素群内に形成されるドットの個数を前記画像データに基づいて前記ド  
10 トの種類毎に決定するドット個数決定手段と、

前記画素群毎に前記各種ドットについて決定したドット個数のデータを、前記制御データとして前記画像出力装置に出力する制御データ出力手段と  
を備えることを要旨とする。

#### 【0035】

15 また、上述した画像処理装置に対応する本発明の画像処理方法は、

単位面積当たりの濃度の異なる各種ドットを形成して画像を出力する画像出力装置に供給されて、該出力装置内で該各種ドットの形成を制御するために用いられる制御データを、該画像を表す画像データに所定の画像処理を施して生成する画像処理方法であって、

20 前記画像を構成する複数の画素が所定の複数個ずつまとめられた画素群について、該画素群内に形成されるドットの個数を前記画像データに基づいて前記ドットの種類毎に決定する工程（a）と、

前記画素群毎に前記各種ドットについて決定したドット個数のデータを、前記制御データとして前記画像出力装置に出力する工程（b）と

25 を備えることを要旨とする。

#### 【0036】

かかる画像処理装置および画像処理方法においては、複数の画素から構成される画素群毎に、該画素群内に形成すべき各種ドットの個数を決定する。そして、

決定した各種ドット個数のデータを画像出力装置に出力する。

【0037】

前述したように、画素群内に形成するドットの個数は、画素群内の各画素についてのドット形成の有無よりも僅かなデータ量で表すことができる。従って、画像データを、画素群のドット個数のデータに変換してから出力すれば、制御データを画像出力装置に迅速に出力することができ、延いては、迅速に画像を出力させることが可能となるので好ましい。

【0038】

また、こうした画像処理装置においては、各種ドットについてのドット個数のデータを符号に変換してから、画像出力装置に出力することとしても良い。

【0039】

制御データは、各種ドットの個数のデータそのままよりも、符号に変換した方がデータ量が小さくなる。従って、各種ドットの個数のデータを一旦、符号に変換してから制御データとして画像出力装置に出力してやれば、制御データを迅速に出力することができ、延いては画像を迅速に出力させることが可能となるので好ましい。

【0040】

また、こうした画像処理装置においては、単位面積当たりの濃度の異なる複数種類のドットについて、画素群内に形成するドットの個数を決定するに際して、次のようにして決定しても良い。すなわち、画素群を構成する画素数に相当する複数個の閾値からなる閾値組を記憶しておき、第1のドット密度データと第2のドット密度データとを、画像データに基づいて画素群毎に生成する。ここで、第1のドット密度データとは、各種ドットの中で単位面積当たりの濃度の最も高い第1のドットが画素群内に形成される密度に対応するデータである。また、第2のドット密度データとは、各種ドットの中で単位面積当たりの濃度が2番目に高い第2のドットまたは前記第1のドットのいずれかのドットが、画素群内に形成される密度に対応するデータである。そして、該閾値組に含まれる閾値の中で、該第1のドット密度データよりも小さな閾値の個数を、その画素群内に形成され

第1のドットの個数と決定する。また、該閾値組に含まれる閾値の中で第2のドット密度データよりも小さな閾値の個数を求め、求めた個数と該第1のドットの個数とに基づいて、その画素群内に形成される第2のドットの個数を決定する。第2のドット密度データを閾値と比較するに際しては、第1のドット密度データより大きな閾値とだけ比較することにより、該第2のドット密度データより小さな閾値の個数を求めてやる。

#### 【0041】

第2のドット密度データは、第1のドットが形成される密度に対応する第1のドット密度データよりも大きな値となるので、第1のドット密度データよりも小さな閾値であれば、比較するまでもなく第2のドット密度データより小さいことは明らかである。従って、第1のドットの個数を決定した後に第2のドットの個数を決定することとして、第2のドット密度データは、第1のドット密度データより大きな閾値とだけ比較することとすれば、第2のドットの個数を迅速に決定することが可能となるので好ましい。

#### 15 【0042】

更に本発明は、上述した画像出力制御方法、画像出力方法あるいは画像処理方法を実現するためのプログラムをコンピュータに読み込ませ、コンピュータを用いて実現することも可能である。従って、本発明はプログラム、あるいは該プログラムを記録した記録媒体としての態様も含んでいる。

#### 20 【0043】

こうしたプログラム、あるいは記録媒体に記録されているプログラムをコンピュータに読み込ませ、該コンピュータを用いて上述の各種機能を実現すれば、例えば画素数の多い画像であっても、迅速に画像を出力させることが可能となる。

#### 25 図面の簡単な説明

図1は、印刷システムを例にとって発明の概要を示した説明図である。

図2は、本実施例の画像処理装置としてのコンピュータの構成を示す説明図である。

図3は、本実施例の画像出力装置としてのプリンタの概略構成図である。

図4は、インク吐出用ヘッドにおけるノズル $N_z$ の配列を示す説明図である。

図5は、第1実施例の画像処理装置で行われる画像印刷処理の流れを示すフローチャートである。

5 図6(a)、図6(b)は、画像データの解像度変換を変換している様子を示す説明図である。

図7は、第1実施例の個数データ生成処理の流れを示すフローチャートである。

図8は、ディザマトリックスの一部を拡大して例示した説明図である。

10 図9は、ディザマトリックスを参照しながら画素毎にドット形成の有無を判断する様子を示す説明図である。

図10(a)、図10(b)は、画素群毎にドット個数のデータが求められている様子を概念的に示した説明図である。

15 図11は、第1実施例の画素位置決定処理の流れを示すフローチャートである。

図12(a)ないし図12(d)は、第1実施例の画素位置決定処理中でドット個数のデータに基づいて画素位置を決定する処理の様子を示す説明図である。

20 図13(a)ないし図13(c)は、第1の変形例の個数データ生成処理中でドット個数を求める様子を示す説明図である。

図14(a)ないし図14(c)は、第2の変形例の画素位置決定処理中でドット個数のデータに基づいて画素位置を決定する様子を示す説明図である。

図15は、第2実施例の個数データ生成処理の流れを示すフローチャートである。

25 図16(a)、図16(b)は、画素群内でのエッジの有無に応じて、画素群内の各画素の階調値を平均階調値で置き換える様子を示した説明図である。

図17(a)ないし図17(c)は、第2実施例においてコンピュータが出力するデータのデータ形式を示す説明図である。

図18は、第2実施例においてコンピュータが出力するデータ形式の他の態様を示す説明図である。

図19は、第2実施例の画素位置決定処理の流れを示すフローチャートである。

5 図20は、第3実施例の画像印刷処理の流れを示すフローチャートである。

図21は、第3実施例の大中小ドット密度データ変換処理中で参照される変換テーブルを概念的示した説明図である。

図22(a)、図22(b)は、第3実施例の個数データ生成処理中で、ドット密度データからドット個数のデータを生成する様子を示した説明図である。

10 図23は、第3実施例の個数データ生成処理の流れを示したフローチャートである。

図24は、ディザマトリックスを参照しながら、各種ドットを形成する画素位置を決定する様子を示した説明図である。

15 図25は、各種ドットの個数のデータをコード化するために参照される変換テーブルを概念的に示した説明図である。

図26は、第3実施例の個数データ生成処理中で、画素位置を特定することなくドットの個数を求める様子を概念的に示した説明図である。

図27は、第3実施例の第2の変形例における個数データ生成処理の流れを示すフローチャートである。

20 図28は、大ドットの個数を決定する処理の流れを示したフローチャートである。

図29は、第3実施例の第2の変形例において中ドットの個数を決定する処理の流れを示したフローチャートである。

25 図30は、第3実施例の第2の変形例において小ドットの個数を決定する処理の流れを示したフローチャートである。

図31は、第3実施例の画素位置決定処理の流れを示したフローチャートである。

図32は、第3実施例の画素位置決定処理中で、ドット個数のデータから画



素位置を決定する様子を示した説明図である。

図33は、第3実施例の画素位置決定処理中で参照される復号化テーブルを概念的に表した説明図である。

図34は、第1の変形例の画像印刷処理中で行われる個数データ生成処理の  
5 流れを示すフローチャートである。

図35は、画素群内の平均階調値に基づいてドット個数が一義的に決定される様子を例示した説明図である。

図36は、第2の変形例の画像印刷処理中で行われる個数データ生成処理の流れを示すフローチャートである。

10 図37は、第3の変形例の画像印刷処理中で行われる画素位置決定処理の流れを示すフローチャートである。

図38は、画素群内の画素の序列が複数通り記憶されている様子を模式的に示した説明図である。

#### 15 発明を実施するための最良の形態

##### 【0044】

本発明の作用・効果をより明確に説明するために、本発明の実施の形態を、次のような順序に従って以下に説明する。

A. 実施形態の概要：

20 B. 第1実施例：

B-1. 装置構成：

B-2. 画像印刷処理の概要：

B-3. 第1実施例の個数データ生成処理：

B-4. 第1実施例の画素位置決定処理：

25 B-5. 第1実施例の変形例：

C. 第2実施例：

C-1. 第2実施例の個数データ生成処理：

C-2. 第2実施例の画素位置決定処理：

D. 第3実施例：

D-1. 第3実施例の画像印刷処理の概要：

D-2. 第3実施例の個数データ生成処理：

D-3. 第3実施例の画素位置決定処理：

5 D-4. 第3実施例の変形例：

E. その他の変形例：

【0045】

A. 発明の概要：

実施例についての詳細な説明に入る前に、図1を参照しながら、本発明の実施形態の概要について説明しておく。図1は、印刷システムを例にとって、実施形態の概要を説明するための説明図である。画像出力制御システムとしての印刷システムは、画像処理装置としてのコンピュータ10と、画像出力装置としてのプリンタ20等から構成されており、コンピュータ10に所定のプログラムがロードされて実行されると、コンピュータ10およびプリンタ20などが全体として  
10 一体の印刷システムとして機能する。プリンタ20は、印刷媒体上にドットを形成することによって画像を印刷する。コンピュータ10は、印刷しようとする画像の画像データに所定の画像処理を施すことによって、プリンタ20が画素毎にドットの形成を制御するためのデータを生成して、該プリンタ20に供給する。  
15

【0046】

20 ここで、通常の印刷システムでは、コンピュータは画像データを、画像を構成する画素毎にドットの形成有無を表したデータに変換してプリンタに供給し、このデータに基づいてドットを形成することにより、画像を印刷している。ここで、印刷しようとする画像の画素数が多くなると、それに伴って、画素毎にドットの形成有無を表したデータのデータ量も増加するので、コンピュータからプリンタ  
25 にデータを出力するために長い時間が必要となり、延いては印刷に要する時間も増加してしまう。こうした点を考慮して、図1に示したコンピュータ10には、ドット個数決定モジュールと個数データ生成モジュールとが設けられており、次のような処理を行う。

#### 【0047】

ドット個数決定モジュールは、画像を構成する複数の画素が所定の複数個ずつまとめられた画素群について、該画素群内に形成されるドットの個数を、画像データに基づいて決定する。ここで画素群毎の個数データは、画像データを画素群  
5 にまとめた後、画素群毎にドット形成の有無を判断することによって生成することができる。あるいは、初めに画像をドット形成の有無による表現形式に変換した後、画素を所定の複数個ずつまとめて画素群を生成して、各画素群内に形成されるドットの個数を決定しても良い。尚、画素群としてまとめられる複数の画素は、必ずしも、互いに隣接した画素でなくても構わない。また、個数データ生成  
10 モジュールは、画素群毎に決定されたドット個数のデータを、プリンタ20に向かって出力する。

#### 【0048】

図1に示したプリンタ20には、序列記憶モジュールと、画素位置決定モジュールと、ドット形成モジュールとが設けられている。序列記憶モジュールには、  
15 画素群内で各画素にドットが形成される画素の序列が、複数通り記憶されている。画素位置決定モジュールは、序列記憶モジュールを参照しながら、画素群毎にドットを形成すべき画素位置を次のようにして決定する。先ず、画素群1つ分の個数データを受け取ると、序列記憶モジュールに記憶された複数の序列の中から序列を1つ選択する。そして選択した序列に従って、画素群を構成する複数の画素  
20 の中から、個数データによって指示された個数の画素を、ドットを形成すべき画素として選択する。こうして画素位置決定モジュールは、個数データを受け取ると、画素群毎に序列を選択して、ドットを形成する画素位置を決定していく。ドット形成モジュールは、こうして決定された画素にドットを形成する。その結果、印刷媒体上に画像が形成されることになる。

#### 【0049】

25 このように、図1に示した印刷システムでは、コンピュータ10からプリンタ20に向かって、ドット形成の有無を表すデータを供給する際に、画素毎のデータではなく、画素群毎に形成されるドットの個数を表すデータを供給する。こう

すれば、個々の画素についてのドット形成の有無を供給する場合よりも、供給するデータ量を少なくすることができる。このため、例え、画像に含まれる画素数が増加した場合でも、コンピュータ10からプリンタ20にデータを迅速に供給することができ、延いては、迅速に画像を印刷することが可能となる。以下では、  
5 こうした印刷システムを例にとって、各種実施例について詳細に説明する。

#### 【0050】

B. 第1実施例：

B-1. 装置構成：

図2は、本実施例の画像処理装置としてのコンピュータ100の構成を示す説明図である。コンピュータ100は、CPU102を中心に、ROM104やRAM106などを、バス116で互いに接続して構成された周知のコンピュータである。  
10

#### 【0051】

コンピュータ100には、フレキシブルディスク124やコンパクトディスク126等のデータを読み込むためのディスクコントローラDDC109や、周辺機器とデータの授受を行うための周辺機器インターフェースPIF108、CRT114を駆動するためのビデオインターフェースVIF112等が接続されている。PIF108には、後述するカラープリンタ200や、ハードディスク118等が接続されている。また、デジタルカメラ120や、カラースキャナ122等をPIF108に接続すれば、デジタルカメラ120やカラースキャナ122で取り込んだ画像を印刷することも可能である。また、ネットワークインターフェースカードNIC110を装着すれば、コンピュータ100を通信回線300に接続して、通信回線に接続された記憶装置310に記憶されているデータを取得することもできる。  
20

#### 25 【0052】

図3は、本実施例のカラープリンタ200の概略構成を示す説明図である。カラープリンタ200はシアン、マゼンタ、イエロ、ブラックの4色インクのドットを形成可能なインクジェットプリンタである。もちろん、これら4色のインク

に加えて、染料濃度の低いシアン（淡シアン）インクと染料濃度の低いマゼンタ（淡マゼンタ）インクとを含めた合計6色のインクドットを形成可能なインクジェットプリンタを用いることもできる。尚、以下では場合によって、シアンインク、マゼンタインク、イエロインク、ブラックインク、淡シアンインク、短マゼンタインクのそれぞれを、Cインク、Mインク、Yインク、Kインク、LCインク、LMインクと略称することがあるものとする。

#### 【0053】

カラープリンタ200は、図示するように、キャリッジ240に搭載された印字ヘッド241を駆動してインクの吐出およびドット形成を行う機構と、このキャリッジ240をキャリッジモータ230によってプラテン236の軸方向に往復動させる機構と、紙送りモータ235によって印刷用紙Pを搬送する機構と、ドットの形成やキャリッジ240の移動および印刷用紙の搬送を制御する制御回路260とから構成されている。

#### 【0054】

キャリッジ240には、Kインクを収納するインクカートリッジ242と、Cインク、Mインク、Yインクの各種インクを収納するインクカートリッジ243とが装着されている。キャリッジ240にインクカートリッジ242、243を装着すると、カートリッジ内の各インクは図示しない導入管を通じて、印字ヘッド241の下面に設けられた各色毎のインク吐出用ヘッド244ないし247に供給される。

#### 【0055】

図4は、インク吐出用ヘッド244ないし247におけるインクジェットノズルNzの配列を示す説明図である。図示するように、インク吐出用ヘッドの底面には、C、M、Y、Kの各色のインクを吐出する4組のノズル列が形成されており、1組のノズル列あたり48個のノズルNzが、一定のノズルピッチkで配列されている。

#### 【0056】

制御回路260は、CPUや、ROM、RAM等がバスで相互に接続された構

成となっている。制御回路260は、キャリッジモータ230および紙送りモータ235の動作を制御することによってキャリッジ240の主走査動作および副走査動作を制御するとともに、コンピュータ100から供給される印刷データに基づいて、各ノズルから適切なタイミングでインク滴を吐出する制御を行う。5  
うして、制御回路260の制御の下、印刷媒体上の適切な位置に各色のインクドットを形成することによって、カラープリンタ200はカラー画像を印刷することができる。

#### 【0057】

尚、各色のインク吐出ヘッドからインク滴を吐出する方法には、種々の方法を適用することができる。すなわち、 piezo素子を用いてインクを吐出する方式や、10  
インク通路に配置したヒータでインク通路内に泡（バブル）を発生させてインク滴を吐出する方法などを用いることができる。また、インクを吐出する代わりに、熱転写などの現象を利用して印刷用紙上にインクドットを形成する方式や、静電気を利用して各色のトナー粉を印刷媒体上に付着させる方式のプリンタを使用す15  
ることも可能である。

#### 【0058】

以上のようなハードウェア構成を有するカラープリンタ200は、キャリッジモータ230を駆動することによって、各色のインク吐出用ヘッド244ないし247を印刷用紙Pに対して主走査方向に移動させ、また紙送りモータ235を20  
駆動することによって、印刷用紙Pを副走査方向に移動させる。制御回路260は、印刷データに従って、キャリッジ240の主走査および副走査を繰り返しながら、適切なタイミングでノズルを駆動してインク滴を吐出することによって、カラープリンタ200は印刷用紙上にカラー画像を印刷している。

#### 【0059】

25 B-2. 画像印刷処理の概要：

図5は、本実施例のコンピュータ100およびプリンタ200が、画像データに所定の画像処理を加えて、印刷媒体上に画像を印刷する処理の流れを示すフローチャートである。かかる画像印刷処理は、後述するように前半部分はコンピュ

ータ１００に内蔵されたＣＰＵの機能を利用して、また、処理の後半部分はプリンタ２００の制御回路２６０に内蔵されたＣＰＵの機能を利用して実行される。以下、図５に従って、本実施例の画像印刷処理について説明する。

#### 【００６０】

- ５ コンピュータ１００は、画像印刷処理を開始すると、先ず初めに、変換すべき画像データの読み込みを開始する（ステップＳ１００）。ここでは、画像データはＲＧＢカラー画像データであるものとして説明するが、カラー画像データに限らず、モノクロ画像データについての同様に適用することができる。

#### 【００６１】

- １０ カラー画像データの読み込みに続いて、色変換処理を行う（ステップＳ１０２）。色変換処理とは、Ｒ、Ｇ、Ｂの階調値の組み合わせによって表現されているＲＧＢカラー画像データを、印刷のために使用される各色の階調値の組合せによって表現された画像データに変換する処理である。前述したように、プリンタ２０はＣ、Ｍ、Ｙ、Ｋの４色のインクを用いて画像を印刷している。そこで、本実施例
- １５ の色変換処理ではＲＧＢ各色によって表現された画像データを、Ｃ、Ｍ、Ｙ、Ｋの各色の階調値によって表現されたデータに変換する。色変換処理は、色変換テーブル（ＬＵＴ）と呼ばれる３次元の数表を参照することで行う。ＬＵＴには、ＲＧＢカラー画像データに対して、色変換によって得られるＣ、Ｍ、Ｙ、Ｋ各色の階調値が予め記憶されているので、このＬＵＴを参照しながら変換すれば、迅速に色変換することが可能である。
- ２０

#### 【００６２】

- 色変換処理を終了すると、解像度変換処理を開始する（ステップＳ１０４）。解像度変換処理とは、画像データの解像度を、プリンタ２００が印刷を行う解像度（印刷解像度）に変換する処理である。前述したように、印刷画質を向上させるためには、画素の大きさを小さくして、より高い解像度で印刷することが効果的である。しかし、印刷解像度を高くすることに合わせて、必ずしも、元の画像データの解像度も高くする必要があるわけではない。何故なら、ドットを形成して画像を印刷する場合は、個々の画素ではドットを形成するか否かの２通りしか
- ２５

取り得ず、例えドットの大きさなどを変えたとしても、画素あたりに表現可能な階調数は高々数階調に過ぎない。これに対して、読み込む画像データは、仮に1バイトのデータとしても、画素あたりに256階調を表現することができる。このように、画素あたりに表現可能な階調が大きく異なっているので、印刷解像度を、読み込む画像データの解像度よりも高解像度に設定することで、印刷画質を向上させることが可能である。このような理由から、図5のステップS104では、画像データの解像度を、より高解像度の印刷解像度に変換する処理を行うのである。尚、より大きな画像を印刷しようとする場合には、解像度を保ったまま画素数を増加させる処理が行われることがある。このような処理も、1つの画素から同じ画像データを有する複数の画素を生成している点で、上述した解像度変換処理と全く同様である。以下の説明では、画像の解像度を高くする場合について説明するが、解像度を高くする処理を画素数を増加させる処理と読み替えてやれば、大きな画像を印刷する場合にも以下の説明を適用することが可能である。

#### 【0063】

図6は、第1実施例において行われる解像度変換の様子を示す説明図である。尚、前述したように色変換によって、C、M、Y、Kの各色毎の画像データが得られるが、以降に説明する処理は、これら各色の画像データのいずれに対しても同様に行われる。そこで、説明の煩雑化を避けるために、以下では色を特定せずに説明する。

#### 【0064】

図6(a)は、色変換後の画像データの一部を拡大して模式的に表したものである。図示するように、画像データは、格子状に配列された画素の各々に階調値が割り当てられたデータである。図6(a)中に示した複数の矩形は、それぞれが画素を模式的に表しており、矩形の中に表示された数値は、各画素に割り当てられた階調値を表している。こうした画像データの解像度をより高い解像度に変換するためには、画素間で補間演算を行うことによって新たな画素を生成しても良いが、本実施例では最も簡便な手法として、画素をより小さな画素に分割することで解像度変換を行う。



#### 【0065】

図6 (b) は、画素を分割することで解像度を交換している様子を示す説明図である。図示した例では、それぞれの画素を、主走査方向（図上で左右方向）に4分割し、副走査方向（図上で上下方向）に2分割することで、1つの画素を8つの画素に分割している。図6 (b) 中に示した破線は、画素が分割されていることを表したものである。こうして生成した小さな画素には、分割前の元の画素の階調値と同じ階調値が割り当てられている。以上のような処理を施すことにより、画像データの解像度は、主走査方向には4倍の解像度に、副走査方向には2倍の解像度に変換されることになる。もちろん、解像度の増加割合は必要に応じて種々の割合に設定することが可能である。

#### 【0066】

以上のようにして解像度を印刷解像度に変換したら、コンピュータ100は、個数データ生成処理を開始する（ステップS106）。ここでは、次のような処理を行う。色変換後の画像データは、画素毎に階調値が割り当てられた階調データである。これに対して、プリンタ200は画素位置に適切な密度でドットを形成することによって画像を印刷する。従って、階調データを、画素毎のドット形成の有無によって表現されたデータに変換した後、プリンタ200に転送する必要がある。また、ドット形成の有無を示すデータを、プリンタ200に画素単位で転送したのでは、画素数が多くなるに従って転送に要する時間が増加してしまうので、画像を迅速に印刷することが困難となる。そこで、本実施例の画像印刷処理では、画素を所定の複数個ずつ画素群としてまとめ、画素群内に形成されるドット個数のデータをプリンタ200に転送している。ここで、画素群内に形成されるドット個数のデータは、予め画像データを、画素毎のドット形成の有無を示すデータに変換した後に、複数の画素を画素群としてまとめることで得ることができる。あるいは、後述するように、初めに複数の画素を画素群にまとめた後、画素群内の各画素に形成されるドットの個数を決定することも可能である。ステップS106の個数データ生成処理では、こうして画素群内で形成されるドット個数のデータを生成して、プリンタ200に転送する処理を行う。個数データ生成

処理の詳細については後述する。

#### 【0067】

プリンタ200の制御回路260に内蔵されたCPUは、コンピュータ100から出力されたドット個数のデータを受け取ると、画素位置決定処理を開始する  
5 (ステップS108)。ここでは、次のような処理を行う。上述したように、コンピュータ100は画素毎にドット形成の有無を示すデータの代わりに、画素群内に形成されるドット個数を示すデータを出力する。そこで、画素位置決定処理では、画素群毎に受け取ったドット個数のデータに基づいて、画素群内でドット  
10 が形成される画素位置を決定する処理を行う。画素位置決定処理の詳細については後述する。

#### 【0068】

以上のようにして、ドットを形成すべき画素位置を決定したら、決定した画素位置にドットを形成する処理を行う(ステップS110)。すなわち、図3を用いて説明したように、キャリッジ240の主走査および副走査を繰り返しながら  
15 インク吐出用ヘッドを駆動してインク滴を吐出することにより、印刷用紙上にインクのドットを形成する。こうしてドットを形成することにより、画像データに対応した画像が印刷されることになる。

#### 【0069】

B-3. 第1実施例の個数データ生成処理:

20 図7は、第1実施例の個数データ生成処理の流れを示すフローチャートである。以下では、フローチャートを参照しながら、個数データ生成処理の内容について詳細に説明する。

#### 【0070】

個数データ生成処理を開始すると、先ず初めに所定の複数個の画素をまとめて  
25 画素群を生成する(ステップS200)。ここでは、解像度変換処理において1つの画素を8つの画素に分割していることから、同一の画素を分割して得られた8つの画素を画素群としてまとめることとする。例えば、図6(a)中で左上隅の画素は、図6(b)中では、左上方にある縦2列横4列の8つ画素に分割され

ているから、これらの画素をまとめて画素群を生成する。尚、画素群としてまとめる画素は、互いに隣接する画素である必要はなく、所定の位置関係にあればどのような画素でも画素群としてまとめることができる。

#### 【0071】

- 5     また、このように同一の画素から分割された画素を画素群としてまとめる場合は、図5の解像度変換処理（図6参照）を省略することも可能である。この場合は、以下の説明中で、「画素群」とある部分を、「解像度変換を行う前の画素」と読み替えることにより、ほぼ同様な処理を行うことができる。

#### 【0072】

- 10    次いで、画素群としてまとめた画素の中から、ドット形成の有無を判断するための着目する画素（着目画素）を1つ設定する（ステップS202）。そして、着目画素に割り当てられた階調値とディザマトリックスの閾値とを比較することにより、着目画素についてのドット形成の有無を判断する（ステップS204）。ディザマトリックスとは、複数の閾値が格子状に記憶された2次元の数表である。
- 15    ディザマトリックスを用いてドット形成の有無を判断する処理について、図8および図9を参照しながら説明する。図8は、ディザマトリックスの一部を例示した説明図である。図示したマトリックスには、縦横それぞれ64画素、合計4096個の画素に、階調値0～255の範囲から万遍なく選択された閾値がランダムに記憶されている。ここで、閾値の階調値が0～255の範囲から選択されて
- 20    いるのは、本実施例では画像データが1バイトデータであり、画素に割り当てられる階調値が0～255の値を取り得ることに対応するものである。尚、ディザマトリックスの大きさは、図8に例示したように縦横64画素分に限られるものではなく、縦と横の画素数が異なるものも含めて種々の大きさとすることができる。

#### 25     【0073】

図9は、ディザマトリックスを参照しながら、着目画素についてのドット形成の有無を判断している様子を概念的に示した説明図である。ドット形成有無の判断に際しては、先ず、着目画素の階調値とディザマトリックス中の対応する位置

に記憶されている閾値と比較する。図中に示した細い破線の矢印は、着目画素の階調値を、ディザマトリックス中の対応する位置に記憶されている閾値と比較していることを模式的に表したものである。そして、ディザマトリックスの閾値よりも着目画素の階調値の方が大きい場合には、その画素にはドットを形成すると判断する。逆に、ディザマトリックスの閾値の方が大きい場合には、その画素にはドットを形成しないと判断する。再び図9に即して説明すると、画像データの左上隅の画素については、画像データの階調値は97であり、ディザマトリックスの閾値は1であるから、この画素にはドットを形成すると判断する。図9中に実線で示した矢印は、この画素にはドットを形成すると判断して、判断結果をメモリに書き込んでいる様子を模式的に表したものである。一方、この画素の右隣の画素については、画像データの階調値は97、ディザマトリックスの閾値は177であり、閾値の方が大きいので、この画素についてはドットは形成しないと判断する。図7のステップS204では、こうしてディザマトリックスを参照しながら、着目画素にドットを形成するか否かを判断する処理を行う。

15       【0074】

次いで、画素群内の全ての画素について以上のような処理を行ったか否かを判断し（ステップS206）、画素群中に未処理の画素が残っている場合は（ステップS206：no）、ステップS202に戻って続く一連の処理を行う。こうして画素群内の全ての画素について、ドット形成有無の判断を終了したら（ステップS206：yes）、画素群内に形成するドットの個数を検出し、画素群に対応付けられた状態でメモリに記憶する（ステップS208）。図9に示した例では、画像の左上隅の画素群については、3つの画素にドットを形成すると判断されているから、この画素群についてはドット個数が「3」である旨を記憶する。

          【0075】

25       以上のようにして、1つの画素群についての処理を終了したら、全画素について処理を終了したか否かを判断し（ステップS210）、未処理の画素が残っていれば、ステップS200に戻って新たな画素群を生成した後、続く一連の処理を行って、その画素群に形成されるドットの個数を記憶する（ステップS208）。

こうした処理を繰り返して行くことにより、画像中の全ての画素についての処理を終了したら（ステップS210：yes）、画素群毎に記憶しておいたドット個数をプリンタ200に向かって出力して（ステップS212）、図7に示した個数データ生成処理を終了する。

5           【0076】

図10（a）は、画像データに上述した個数データ生成処理を施すことによって得られるデータを、概念的に表した説明図である。図中に示した複数の矩形は、それぞれ画素群を表しており、画素群内に表示された数値は、該画素群に形成されるドットの個数が記憶されている様子を表している。本実施例では、コンピュータ100は、色変換後の画像データを図10（a）に示すようなデータに変換した後、画素群毎に記憶された個数のデータのみをプリンタ200に向かって出力する。このように個数のデータのみを出力すれば、画素毎にドットの形成有無を示すデータを出力する場合よりも、データ量が減少するので迅速に出力することが可能となる。この点につき、補足して説明する。

15           【0077】

図10（b）は、画素群内の各画素について、ドット形成の有無を判断した様子を示す説明図である。図10（b）中に示した細い破線は、画素群が複数の画素から構成されていることを示しており、画素に付された斜線は、その画素にはドットを形成すると判断されていることを示している。

20           【0078】

今、コンピュータ100からプリンタ200に対して、図10（b）に示した状態のデータ、すなわち画素毎にドット形成の有無を示すデータを出力するものとする。ドットの種類は1種類であるとすれば、各画素はドットが形成されるか、形成されないかのいずれかの状態しか取り得ないから、1画素あたりのデータ量は1ビットで足りる。画素群は8つの画素で構成されているから、プリンタ200に出力すべきデータ量は、画素群あたり8ビットとなる。

          【0079】

これに対して、本実施例のように画素群に形成されるドットの個数を出力する

場合は、1つの画素群内に形成されるドットの個数は0～8の値しか取り得ないから、画素群あたり4ビットあればドット個数を表現することができる。すなわち、画素毎にドット形成の有無を示すデータを出力する場合に比べて、データ量を半減させることができる。このため、画素群毎のドット個数を出力することによって、プリンタ200に迅速にデータを出力することが可能となるのである。こうしてコンピュータ100から転送されたドット個数のデータは、プリンタ200側で次に説明する画素位置決定処理が施されて、画素毎にドットの形成有無を示すデータに変換される。以下、画素位置決定処理について説明する。

#### 【0080】

#### 10 B-4. 第1実施例の画素位置決定処理：

図11は、第1実施例の画像印刷処理中で行われる画素位置決定処理の流れを示すフローチャートである。かかる処理は、プリンタ200の制御回路260に内蔵されたCPUが、コンピュータ100から転送されてきた画素群毎のドット個数  
15 数  
を示すデータを受け取って実行する。また、図12は、画素群毎のドット個数  
を示すデータが、画素位置決定処理を施されることによって、画素毎のドット形成の有無を示すデータに変換される様子を概念的に示した説明図である。以下、図11および図12を参照しながら、画素位置決定処理の内容について説明する。

#### 【0081】

画素位置決定処理を開始すると、先ず初めに、処理の対象とする画素群を1つ  
20 選択し（図11のステップS300）、その画素群に形成されるドット個数を取  
得する（ステップS302）。図12（a）には、コンピュータ100から画素  
群毎に転送されてきたドット個数  
を示すデータが概念的に示されている。画素位置  
を決定する画素群として、ここでは、図中で左上隅にある画素群を選択したも  
のとする。図11のステップS302では、選択した画素群に形成されるドット  
25 個数として「3」を取得する。

#### 【0082】

次いで、画素群内の各画素にドットが形成される序列を参照することにより、ドットを形成する画素を決定する処理を行う（ステップS304）。ここでは、

ディザマトリックスを、ドットが形成される序列として読み替えて利用する。図 8 を用いて説明したように、ディザマトリックスの各画素には閾値が設定されている。また、ある画素にドット形成の有無を判断する際には、画像データの階調値とディザマトリックスの閾値とを比較して、階調値の方が大きければ、その画素にはドットを形成すると判断する。すなわち、ディザマトリックスの閾値が小さい画素ほどドットが形成され易くなることから、ディザマトリックスはドットが形成される画素の序列を表していると考えることができる。本実施例では、ディザマトリックスの有するこうした性質に着目して、画素群内の各画素の序列としてディザマトリックスを利用するのである。

#### 10       【0083】

再び図 12 を参照しながら、詳しく説明する。ここでは、対象とする画素群が図 12 (a) 中の左上隅の画素群であるとしているから、ディザマトリックス上で、この画素群の各画素に対応する位置に記憶されている閾値を取得する。図 12 (b) は、図 8 に示したディザマトリックスから、対応する画素位置の閾値が読み出された様子を模式的に示したものである。こうして読み出した閾値が小さい画素から順番に、ドットが形成されることになる。図 12 (a) に示すように、処理中の画素群には形成されるドット個数は 3 個だから、各画素にドットが形成される序列に基づいて、図 12 (c) に示すように画素位置を決定することができる。すなわち、図 12 (c) 中で実線で囲って示した最も閾値の小さい画素と、破線で囲った 2 番目に閾値の小さい画素と、一点鎖線で囲った 3 番目に閾値の小さい画素の 3 つの画素を、ドットが形成される画素として決定することができる。

#### 【0084】

以上のような操作を行うことにより、処理対象として選択した画素群について画素位置を決定したら、全ての画素群についての処理を終了したか否かを判断する (図 11 のステップ S 306)。未処理の画素群が残っていれば (ステップ S 306 : no)、ステップ S 300 に戻って新たな画素群を選択し、続く一連の処理を行う。こうした処理を繰り返すことにより、図 12 (a) に例示の画素群毎にドット個数を示すデータは、図 12 (d) に示すような、画素毎にドット形

成の有無を表すデータに変換されていく。尚、図12(d)中で斜線が付された画素は、ドットが形成される画素を示している。そして、全ての画素群について処理が終了したら(ステップS306: yes)、図11に示した画素位置決定処理を終了して、図5の画像印刷処理に復帰する。

5           【0085】

以上、第1実施例の画像印刷処理および、かかる処理中で行われる個数データ生成処理、画素位置決定処理について詳しく説明した。このように第1実施例の画像印刷処理では、画像処理が施されたデータをコンピュータ100からプリンタ200に向かって転送するに際して、画素毎にドット形成の有無を示すデータの代わりに、画素群に形成されるドット個数を示すデータを転送する。こうすれば、データの転送量を大幅に減少させることができるので、例えば画像に含まれる画素数が増加したとしても、データの転送を速やかに完了して、迅速に画像を印刷することが可能となる。

          【0086】

15       また、上述したように、コンピュータ100側の個数データ生成処理で参照するディザマトリックスを、プリンタ200側の画素位置決定処理で参照するディザマトリックスと同じマトリックスにしておけば、コンピュータ100からプリンタ200にドット個数の情報のみに圧縮して転送した場合でも、画素位置を完全に復元することができる。実際、図10(b)と図12(d)とを比較すれば  
20       明らかなように、コンピュータ100側で画素毎にドット形成有無を判断して得られた画素位置は、プリンタ200側で決定した画素位置と一致しており、画素位置が完全に復元されていることを示している。このため、コンピュータ100からドット個数のデータを迅速に転送しながら、プリンタ200側でドットを形成する画素位置を適切に決定することにより、高画質な画像を迅速に印刷すること  
25       が可能となる。

          【0087】

B-5. 変形例:

上述した第1実施例には種々の変形例が存在する。以下、これら変形例につい



て簡単に説明する。

#### 【0088】

##### (1) 第1の変形例：

図7ないし図10を用いて説明したように、第1実施例の個数データ生成処理  
5 では、画像データを、画素毎にドット形成有無を表すデータに一旦変換し、得ら  
れたデータを、画素群毎のドット個数を示すデータに変換してプリンタ200に  
転送している。すなわち、コンピュータ100内では、画素位置を特定した状態  
でドット形成の有無を判断しているものの、プリンタ200に転送する段階では、  
画素位置に関する情報は省略されてしまい、画素群毎のドット個数の情報のみが  
10 転送される。こうした点に鑑みて、第1の変形例では、個数データ生成処理にお  
いて画素位置を特定することなく、画素群内に形成されるドット個数のみを生成  
する。

#### 【0089】

図13は、第1の変形例において行われる個数データ生成処理の概要を示した  
15 説明図である。図13(a)は、第1の変形例の個数データ生成処理において参  
照される簡易ディザマトリックスを例示した説明図である。第1実施例において  
参照したような通常に用いられるディザマトリックスでは、画素毎に閾値が設定  
されている(図8を参照)。これに対して、第1の変形例で参照する簡易ディザ  
マトリックスでは、閾値が画素毎に対応付けられておらず、画素群毎にひとまと  
20 まりの状態で、換言すれば、複数個の閾値が1組にまとめられて画素群毎に対応  
付けられた状態で記憶されている。また、各画素群に記憶されている閾値の個数  
は、画素群を構成する画素の数と同数となる。図13(a)に示した例では、簡  
易ディザマトリックスの左上隅の画素群には、{1, 42, 58, 109, 17  
0, 177, 212, 255}の8つの値からなる閾値の組が対応付けられて記  
25 憶されている様子が概念的に示されている。同様に、その右隣の画素群には、{5,  
48, 70, 79, 161, 186, 223, 242}の8つの値で構成される  
閾値の組が記憶されている。

#### 【0090】

第1の変形例における個数データ生成処理では、画素群の画像データを、こうした閾値の組と比較することによって、画素位置を特定することなく、画素群内に形成すべきドット個数を決定する。ここで説明の便宜から、処理しようとする画像データが、第1実施例の説明中で用いたものと同じ画像データであったとする（図6参照）。画像データの左上隅の画素群について見ると、画素群内の全ての画素は階調値97となっている。一方、簡易ディザマトリックス上で対応する位置の画素群には、8つの閾値{1, 42, 58, 109, 170, 177, 212, 255}が記憶されている。これら8つの閾値の中で、画素群の階調値97よりも小さな値は{1, 42, 58}の3つあるから、この画素群には3つのドットが形成されると判断する。図13(b)中で閾値の一部が破線で囲われているのは、これら閾値が画素群の階調値よりも小さいことを表したものである。このように、画素群毎に閾値の組を記憶しておき、画素群の階調値と比較してやれば、画素群内での画素位置を特定することなくドット個数を決定することができる。以上の操作を全ての画素群について行えば、図13(c)に示すように、画素群毎にドット個数を決定することができる。

#### 【0091】

また、第1の変形例中で参照した簡易ディザマトリックスは、第1実施例の説明で用いた通常のディザマトリックスに対応したマトリックスとなっている。すなわち、図13の簡易ディザマトリックス中で画素群毎に記憶されている閾値の組は、図8のディザマトリックスの各画素に記憶されている閾値を、画素群毎にまとめて組にしたものとなっている。このように、簡易ディザマトリックスを通常のディザマトリックスに対応したものとしておけば、簡易ディザマトリックスを使用して画素位置を特定せずにドット個数を決定した場合でも、通常のディザマトリックスを用いて画素位置を特定しながら決定したドット個数とを全く同じ結果を得ることができる。実際、同じ画像データに対して、通常のディザマトリックスを用いて得られたドット個数のデータ（図10(a)参照）と、簡易ディザマトリックスを用いて得られたデータ（図13(c)参照）とは完全に一致したとなっている。

#### 【0092】

上述したように第1の変形例では、画素毎に記憶されている閾値の組と、画像データの階調値とを比較するだけで画素群に形成すべきドット個数を求めることができる。すなわち、閾値と画像データの階調値とを画素群内の画素毎に比較する  
5 必要がないので、画素群毎にドット個数を示すデータを迅速に得ることが可能となる。

#### 【0093】

また、第1の変形例では、画像データの階調値よりも小さな閾値の個数のみを求めればよいから、それぞれの組内で閾値を大きさの順に並べて記憶しておけば、  
10 処理速度の更なる高速化を図ることができる。再び図13の例示を参照しながら説明すると、左上隅の画素群は、画像データの階調値97であり、閾値の組は{1, 42, 58, 109, 170, 177, 212, 255}と大きさの順に記憶されている。画像データの階調値は0~255の値を取り得るから、階調値97は比較的小さな値と言える。そこで、この階調値を小さい閾値から順に大きさを比  
15 較していく。すなわち、初めは一番小さな閾値「1」と比較する。当然、階調値「97」の方が大きいので、今度は、隣に記憶されている1つ大きな閾値「42」と比較する。ここでも階調値「97」の方が大きいから、その隣の閾値「58」と比較する。こうして、階調値を小さい側の閾値から順番に比較していく。そして、閾値「109」と比較する段階で、初めて階調値「97」の方が小さいと判  
20 断される。組内の閾値は大きさの順に記憶されていることから、一旦、階調値の方が小さいと判断されたら、それ以降の閾値についても階調値の方が小さいことは明らかであり、もはや一々比較する必要はない。従って、各画素群には8つの閾値が記憶されているにも関わらず、この画素群については4つの閾値と比較するだけでドット個数を決定することが可能となる。具体的には、予め所定の閾値  
25 (例えば100)を定めておき、画像データの階調値がこの閾値よりも小さければ、小さな閾値から比較していけばよい。

#### 【0094】

もちろん、画像データの階調値が比較的大きな値である場合は、大きい側の閾

値から比較していけばよい。例えば、上述の説明で仮に階調値が「200」であるとすれば、最も大きな閾値「255」から初めて、その隣の閾値「212」、更にその隣の閾値「177」の3つの閾値と比較するだけで、その画素群に形成すべきドット個数を決定することができる。具体的には、予め所定の閾値（例えば155）を定めておき、画像データの階調値がこの閾値よりも大きければ、大きな閾値から比較していけばよい。

#### 【0095】

あるいは、画像データの階調値と比較する閾値を、直前に処理した画素群での処理結果に基づいて選択することとしても良い。再び図13の例示を参照しながら説明する。左上隅の画素群については、画像データの階調値より小さな閾値は3つ（すなわち、ドット個数は3個）である。そこで、該画素群の右隣の画素群については、先ず初めに、閾値の組の中で3番目に小さな閾値と画像データの階調値とを比較する。すなわち右隣の画素群に設定された8つの閾値{5, 48, 70, 79, 161, 186, 223, 242}の中で3番目に小さな閾値「70」であるから、先ず初めにこの閾値と画像データの階調値「102」とを比較するのである。この場合は、画像データの階調値の方が閾値よりも大きいので、次は4番目に小さな閾値「79」、更にその次は5番目に小さな閾値「161」と比較する。5番目に小さな閾値「161」は画像データの階調値「102」よりも大きくなるので、この画素群についてはドット個数は4個であると決定することができる。結果として、画像データの階調値と閾値とを3回比較するだけでドット個数が決定できたことになる。尚、ここでは、先に処理した画素群のドット個数と同じ順番の3番目の閾値から比較を始めたが、必ずしも同じ順番である必要はなく、例えば、1つ大きな4番目の閾値から比較を開始したり、あるいは1つ小さな2番目の閾値から比較を開始することとしても良い。

#### 【0096】

一般に、画像データの階調値は通常、緩やかに変化することが多いので、隣接する画素群間では、画素群内に形成されるドット個数も大きくは変わらないことが多い。従って、上述したように、画像データの階調値と比較する閾値を、先に

処理した画素群のドット個数に基づいて選択してやれば、少ない比較回数でドット個数を決定することが可能となる。

【0097】

以上に説明したように、画素群に対応付けられた閾値を大きさの順に記憶しておけば、画素群内に形成すべきドット個数を迅速に決定することが可能となるのである。

【0098】

(2) 第2の変形例：

前述した第1実施例の画素位置決定処理では、画素群毎のドット個数を示すデータを受け取ると、ディザマトリックスを参照しながら、それぞれの画素群内でドットが形成される画素位置を決定している（図12参照）。しかし、ドット個数からドットが形成される画素位置を決定するためには、必ずしも画素毎の閾値まで分かっている必要はなく、画素群内の各画素についてのドットが形成される画素の序列が分かればよい。こうした点に鑑みて、第2の変形例では、画素位置決定処理においてディザマトリックスの代わりに、画素の序列を記憶したマトリックス（以下では、こうしたマトリックスを序列マトリックスと呼ぶ）を参照しながら画素位置を決定する。

【0099】

図14は、第2の変形例の画素位置決定処理において、序列マトリックスを参照しながら画素位置を決定する様子を概念的に示した説明図である。図14(a)は、序列マトリックスを概念的に示した説明図である。図中で太い実線で囲われた矩形は画素群を表しており、各画素群は細い破線で示されているように、8つの画素に区分されている。また、各画素の中に表示されている数字は、画素群内でドットが形成される序列（換言すれば、ドットが形成される順番）を示している。

【0100】

こうした序列マトリックスを用いれば、ドット個数に基づいて画素位置を簡便に決定することができる。ここでは、ドット個数のデータが図12(a)に示し

たデータである場合を例にとりて、具体的に説明する。尚、このデータは、前述の第1実施例の画素位置決定処理についての説明で用いたものと同じデータである。図12(a)に示したデータによれば、左上隅の画素群に形成されるドット個数は「3」となっている。そこで、図14(a)に示した序列マトリックスの  
5 左上隅の画素群の中から、序列が1番～3番の3つの画素を選択して画素位置を決定する。図14(b)は、こうして3つの画素を選択して画素位置を決定している様子を模式的に示した説明図であり、画素中の数字を囲む実線は、その画素が選択されたことを表している。以上のような操作を全ての画素群に対して繰り返していけば、図14(c)に示すように、ドットを形成する画素位置を全て決  
10 定することができる。尚、図14(c)中で斜線が付された画素は、ドットが形成されることを表している。

#### 【0101】

このように、画素群内で各画素にドットが形成される序列に基づいて画素位置を決定することとすれば、序列の数字がドット個数以下の画素を選択するだけで  
15 画素位置を決定することができる。すなわち、選択した画素の個数を計数しながら、ドット個数に示された数の画素を選択する必要があるないので、画素位置を簡便に決定することが可能となる。

#### 【0102】

また、序列マトリックスに記憶されている数値は、高々、1つの画素群に含ま  
20 れる画素の数までしか取り得ず、ディザマトリックスに記憶される閾値よりも小さな数値である。従って、序列マトリックスはディザマトリックスよりも僅かな容量で記憶することができる。画素位置決定処理が行われるプリンタは、十分な記憶容量が搭載されていないことがあるので、序列マトリックスを用いて画素位置を決定すれば、プリンタの記憶容量を節約することができるという利点も得る  
25 ことができる。

#### 【0103】

尚、図14(a)に示した序列マトリックスは、個数データ生成処理中で画素毎にドット形成の有無を判断するために参照したディザマトリックスと対応した

マトリックスとなっている。すなわち、第1実施例の画素位置決定処理についての説明の中で、ディザマトリックスに設定されている閾値は、ドットが形成される序列を表していると考えることができる旨を説明したが、図14(a)の序列マトリックスに設定されている序列は、画素群内の各画素についてディザマトリックスによって決まる序列と、同じ序列となっている。このように、序列マトリックスを、個数データ生成処理で参照したディザマトリックスに対応したマトリックスとしておけば、序列マトリックスを用いた場合でも、画素位置を適切に決定することができる。実際、図14(c)と図10(b)とを比較すれば明らかに、序列マトリックスを参照して決定した画素位置は、画素毎にドット形成の有無を判断して求めた画素位置と一致しており、画素位置が適切に決定されていることが分かる。

#### 【0104】

また、前述した簡易ディザマトリックス(図13参照)は図8のディザマトリックスを元にして生成され、図14に示した序列マトリックスは図8のディザマトリックスと対応するマトリックスとなっている。このように、同じディザマトリックスを介して、互いに対応付けられた簡易ディザマトリックスおよび序列マトリックスを使用すれば、画素位置を適切に決定することができる。もっとも、これらマトリックスが互いに対応付けられていない場合でも、マトリックスの大きさを揃えておくことが望ましい。ここで、2つのマトリックスの大きさが等しいとは、マトリックス間で行方向および列方向の画素数が等しいことを言う。

#### 【0105】

マトリックスよりも大きな画像を処理する場合は、1つのマトリックスを画像上での位置をずらしながら繰り返し使用することになる。簡易ディザマトリックスおよび序列マトリックスの大きさが違っていると、位置をずらすたびに互いの位置関係が異なってしまうので、処理が不安定となり、結果として画質が悪化するおそれがある。従って、簡易ディザマトリックスおよび序列マトリックスを、同じ大きさのマトリックスとしておけば、互いの位置関係を常に一定に保つことができる。この結果、安定した処理を行うことができ、良好な画質を維持するこ

とができるという利点を得られる。

#### 【0106】

##### C. 第2実施例：

以上に説明した第1実施例では、1つの画素の複数の画素に分割し、同じ画素  
5 から分割された画素同士を画素群としてまとめる場合について説明した。1つの  
画素を複数の画素に分割することは、例えば、入力された画像データよりも高い  
解像度で画像を印刷しようとする場合などに起こり得る。こうした第1実施例に  
おいては、画素群としてまとめられた各画素は同じ階調値を有している。しかし、  
本発明は、画素群内の各画素が異なる階調値を有する場合にも適用することがで  
10 きる。例えば、印刷解像度が入力された画像データの解像度と同じ解像度に設定  
されていて、1つの画素群中に画像データの複数の画素が含まれている場合には、  
画素群内の各画素で階調値が異なる場合が生じ得る。また、画像データよりも高  
い解像度で画像を印刷するために新たな画素を生成する場合でも、新たに生成し  
た画素の階調値を補間演算によって決定した場合には、画素群内の各画素で階調  
15 値は異なった値となる。更には、印刷解像度が画像データの解像度よりも少しだ  
け高いような場合は、例え1つの画素を分割して複数の新たな画素を生成したと  
しても、1つの画素群に異なる画素から生成された画素が含まれることが考えら  
れる。こうした場合にも、画素群内の各画素が同じ階調値を有しているとは限ら  
ない。以下では、第2実施例として、この様な場合に本発明を適用した実施例に  
20 ついて説明する。

#### 【0107】

##### C-1. 第2実施例の個数データ生成処理：

図15は、第2実施例の個数データ生成処理の流れを示すフローチャートであ  
る。かかる処理も前述した第1実施例の個数データ生成処理と同様に、図5に示  
25 した画像印刷処理中でコンピュータ100のCPUによって実行される処理であ  
る。

#### 【0108】

第2実施例の個数データ生成処理においても、先ず初めに所定の複数個の画素



をまとめて画素群を生成する（ステップS400）。ここでは、第1実施例にならって、縦2列横4列の8つの画素をまとめて画素群を生成するものとする。もちろん、画素群としてのまとめ方は、こうしたまとめ方に限定されるものではなく、種々の方法でまとめることができる。

5           【0109】

次いで、画素群内にエッジが含まれているか否かを判断する（ステップS402）。本実施例では、画素群に含まれる複数の画素の中で、最も大きな階調値と最も小さな階調値との階調差が所定値以上の場合に、その画素群にはエッジが含まれているものと判断する。もちろん、エッジの検出方法は、こうした方法に限らず種々の方法を適用することが可能である。

          【0110】

そして、画素群にエッジが含まれていないと判断された場合は（ステップS402：no）、画素群の平均階調値を算出して各画素の階調値を平均階調値で置き換えてやる（ステップS404）。この結果、画素群内の各画素は同じ階調値となるので、前述した第1実施例と同様の方法を用いて、画素群内に形成されるドット個数を決定することができる。

          【0111】

図16は、画素群内でのエッジの有無を判断して、エッジがないと判断した場合には、画素群内の各画素の階調値を平均階調値で置き換える様子を示した説明図である。図16（a）には、画素群内の各画素の階調値を示している。図上で左上隅にある画素群については、最大階調値および最小階調値はそれぞれ階調値100と階調値97であり、その階調差は3と、比較的小さな値となっている。これに対して、その画素群の右下にある画素群については、最大階調値は階調値132、最小階調値は階調値99と、階調差は33もある。そこで適当な閾値（例えば階調値20）を設定しておき、画素群内の階調差が閾値より大きければエッジが含まれていると判断する。逆に、画素群内の階調差が閾値より小さい場合はエッジは含まれていないと判断し、画素群内ので平均階調値を算出して、全ての画素の階調値を平均階調値で置き換えてやる。

### 【0112】

- 図16(b)は、こうして画素群にエッジが含まれていないと判断された場合に、画素群内の各画素の階調値が平均階調値で置き換えられた様子を表している。図示した例では、左上隅の画素群および、その右隣あるいは下の画素群について
- 5 はいずれも階調差が小さく、従って、画素群内の全ての画素が、それぞれ平均階調値99、103、94で置き換えられている。こうして画素の階調値が平均階調値で置き換えられた画素群については、画素群内の各画素は全て同じ階調値となっているので、前述した第1実施例と同様にして画素群に形成すべきドット個数を決定することができる(図15のステップS406)。これに対して、右下
- 10 の画素群については階調差が33と大きく、エッジが含まれていると考えられるので、こうした階調値の置き換えは行わない。こうした画素群については、画素の階調値とディザマトリックスの閾値とを比較しながら画素毎にドット形成の有無を判断して(図9参照)、各画素についての判断結果を記憶しておく(図15のステップS408)。

### 15 【0113】

- 以上のようにして、1つの画素群についての処理を終了したら、画像データ中の全ての画素について処理を終了したか否かを判断する(ステップS410)。そして未処理の画素が残っていれば(ステップS410: no)、ステップS400に戻って新たな画素群を生成し、続く一連の処理を繰り返す。こうして全て
- 20 の画素を画素群としてまとめて上述の処理を終了したら(ステップS410: yes)、エッジを含まない画素群についてはステップS406で記憶しておいたドット個数を、またエッジを含む画素群についてはステップS408で記憶しておいた画素毎にドット形成の有無を示すデータ(ドットデータ)を、プリンタ200に向かって出力する(ステップS412)。この結果、画素群のドット個数のデータと、画素群内の各画素についてのドットデータとが、混在した状態でプ
- 25 リンタ200に出力されることになる。第2実施例の個数データ生成処理では、ドット個数のデータと、各画素についてのドットデータとを区別しながら出力するために、これらデータを次のような形態で出力する。

#### 【0114】

図17は、第2実施例においてコンピュータ100がプリンタ200に向かって出力するデータの形式を示した説明図である。画素群に形成するドット個数を出力する場合は、図17(a)に示すように、画素群あたり4ビットのデータとして出力する。ここでは、1つの画素群は8個の画素で構成されるものとしているから、ドット個数は0～8の値しか取り得ず、4ビットあればドット個数を表現することができる。一方、ドット個数ではなく、各画素についてのドット形成の有無を示すドットデータを出力する場合は、図17(b)に示すような形式でデータを出力する。すなわち、初めの4ビットに9～15のいずれかの値を設定し(図17(b)の例示では「9」が設定されている)、続く8ビット分のデータに、画素毎のドット形成の有無を設定する。ドット個数は0～8の値しか取り得ないから、先頭の4ビットのデータが9以上の値となっている場合は、続く8ビット分のデータはドット個数ではなく画素毎のドット形成の有無を表していると解釈すればよい。そして、それぞれのビットを、例えば図17(c)に示すような順番で各画素に対応付けておけば、画素群内でのドット形成の有無を示すデータを出力することができる。このような図17に示した形式でデータを出力する場合、エッジを含まない画素群については4ビットのデータとしてプリンタ200に転送され、エッジを含む画素群については12ビットのドットデータとして転送されることになる。

#### 【0115】

もちろん、ドット個数のデータと画素毎のドットデータとを混在させて出力する方法は、こうした方法に限定されるものではなく、例えば識別用のビットを付加してもよい。例えば、図18(a)に示すように、識別用のビットが「0」の場合は、続く4ビットのデータがドット個数を表しているものと判断する。また、図18(b)に示すように、識別用のビットが「1」の場合は、続く8ビットのデータが、ドットデータを表しているものと判断しても良い。このようにしてデータを出力した場合は、エッジを含まない画素群については5ビットのデータとしてプリンタ200に転送され、エッジを含む画素群については9ビットのデータとしてプリンタ200に転送されることになる。

タとして転送されることになる。尚、図18では、識別用のビットは、ドット個数のデータ、あるいはドット形成の有無を示すドットデータの先頭に付加されて転送されるものとして説明した。しかし、これに限らず、ドット個数のデータやドットデータとは別に、識別用のビットだけを転送することとしても良い。

5           【0116】

図17の転送方法と図18の転送方法とについてデータの転送量を比較すれば明らかなように、エッジを含まない画素群については、図17の方法による方が、図18に示した方法よりもデータ転送量が少なくなる。しかし、エッジを含む画素群については、逆に、図18の方法による方がデータ転送量が少なくなる。

- 10   このことから、画素群中にエッジの含まれる割合が高い場合は、識別用のビットを用いた図18の方法を用いてデータを転送し、逆に、エッジの含まれる割合が低い場合は、図17に示した方法を用いてデータを転送すればよい。通常は、画素群にエッジが含まれる割合はそれほど高くないので、図17の方法を用いた方が迅速にデータを転送することができる。

15           【0117】

図15のステップS412では、以上に説明したようにして、画素群のドット個数のデータと、画素毎のドット形成の有無を示すドットデータとを混在させた状態でプリンタ200に出力する。こうして、全ての画素群についてのデータを出力したら、図15に示した第2実施例の個数データ生成処理を終了して、画像

- 20   印刷処理に復帰する。

            【0118】

C-2. 第2実施例の画素位置決定処理：

- 上述したように、第2実施例においては、ドット個数のデータと画素毎のドット形成の有無を示すデータとが、コンピュータ100から同時に送られてくるの  
25   で、プリンタ200では次のようにしてドットを形成する画素位置を決定する。  
      尚、以下の説明では、データは図17に示した方法で転送されるものとする。

            【0119】

      図19は、第2実施例において、ドットを形成する画素位置を決定する処理の

流れを示すフローチャートである。処理を開始すると、先ず初めに4ビット分のデータを読み込む(ステップS500)。そして、読み込んだデータが9以上か否かを判断する(ステップS502)。前述したように、ここでは画素群に含まれる画素数は8個であり、ドット個数は0~8の値しか取り得ない。このことから、読み込んだ値が9以上の値であれば、そのデータはドット個数を表しているのではなく、続くデータがドットデータであることを表していると考えることができる。逆に、読み込んだ値が0~8の値であれば、そのデータは、画素群に形成されるドット個数を表すデータと考えることができる。

#### 【0120】

10   そこで、読み込んだデータが9以上か否かを判断し、9以上でない場合は(ステップS502: no)、読み込んだデータをドット個数のデータと解釈し、前述した第1実施例と同様にして、画素群内でドットを形成する画素位置を決定する(ステップS504)。一方、読み込んだデータが9以上である場合は(ステップS502: yes)、続く8ビットのデータを読み込んで、このデータが画  
15   素毎にドット形成の有無を表すものと解釈することにより、ドットを形成する画素位置を決定する(ステップS506)。

#### 【0121】

以上のようにして、画素群1つについて、ドットを形成する画素位置を決定したら、全ての画素群についての処理を終了したか否かを判断する(ステップS  
20   508)。そして、未処理の画素群が残っていればステップS500に戻って続く一連の処理を行う。こうして、全ての画素群について画素位置を決定するまで上述した処理を繰り返し、全ての画素群についての処理が終了したら、図19に示した第2実施例の画素位置決定処理を抜けて、画像印刷処理に復帰する。

#### 【0122】

25   以上、図15ないし図18に示した個数データ生成処理および図19に示した画素位置決定処理を行う第2実施例の画像印刷処理によれば、画素群内で各画素の階調値が同じ値とならない場合でも、エッジを含まない画素群についてはドット個数を示すデータを転送することができるので、プリンタ200に迅速にデー

タを転送して画像を迅速に印刷することが可能となる。

#### 【0123】

- また、第2実施例の画像印刷処理では、エッジを含む画素群については画素毎にドット形成の有無を示すドットデータを転送しているので、エッジを含む画素
- 5 群の割合が高くなるほど、プリンタ200へのデータ転送に要する時間が増加する。このことから、エッジの有無を検出するための判定基準を高くして、エッジが検出され難くするほどデータ転送に要する時間が短くなり、延いては画像の迅速な印刷が可能となる。

#### 【0124】

- 10 その一方で、エッジが検出されなかった画素群については、画素群内の各画素の階調値を平均階調値によって置き換えてしまうので、データ転送時間の短縮化を図る余りエッジの判定基準を高くしすぎると（エッジが検出され難くすると）、印刷画質の低下が懸念される。しかし、高い印刷画質が要求される画像は、通常は印刷解像度が入力される画像データの解像度よりの高い値に設定されている。
- 15 高解像度に変換するために、解像度変換処理（図5のステップS104）で画素を分割して新たな画素を生成した場合でも、あるいは補間演算を行って新たな画素を生成した場合でも、ほとんどの画素群では階調値は緩やかに変化することになる。このことから、高い印刷画質が要求される画像では、ほとんどの場合、エッジの判定基準を高くせずとも大部分の画素群ではエッジは検出されず、高い画
- 20 質を維持したままデータを迅速に転送することができる。

#### 【0125】

- 一方、印刷解像度が入力された画像データの解像度と同程度の低解像度である場合は、画素群にエッジが検出される割合が高くなる。従って、プリンタ200へのデータ転送に要する時間を短縮化するためには、判定基準を高くしてエッジ
- 25 が検出され難くする必要が生じる。しかし、このように印刷解像度が比較的低い値に設定されている場合は、画質よりも迅速な印刷が要求されないことが多いので、エッジの判定基準を高くしても画質の悪化が問題になることはない。

#### 【0126】

D. 第3実施例：

- 以上に説明した各種の実施例は、個々の画素では、ドットを形成した状態あるいは形成しない状態に対応した2つの階調値しか表現し得ないものとして説明した。しかし、プリンタの中には、形成するドットの大きさを変えたり、あるいは
- 5 ドットを形成するために用いるインクの濃さを変えることで、個々の画素単独で、より多くの階調値を表現可能としたプリンタも存在する。本発明は、こうしたいわゆる多値プリンタに対しても有効に適用することができる。以下では、本発明を多値プリンタに適用した第3実施例について説明する。

【0127】

10 D-1. 第3実施例の画像印刷処理の概要：

- 図20は、第3実施例の画像印刷処理の流れを示すフローチャートである。第3実施例の画像印刷処理は、図5を用いて説明した第1実施例の画像印刷処理に対して、色変換処理後のデータを大中小ドットのデータに変換する点が大きく異なっている。以下では、この相違点を中心に、第3実施例の画像印刷処理について説明する。尚、ここでは、プリンタ200はドットの大きさを変更可能な多値
- 15 プリンタであるものとして説明するが、もちろん以下の説明は、ドットの大きさではなくインクの濃度を変更するプリンタや、更にはドットの大きさとインクの濃度とを同時に変更可能な多値プリンタにも、同様に適用することができる。

【0128】

- 20 第3実施例の画像印刷処理においても、第1実施例の画像印刷処理と同様に、先ず初めに、変換すべき画像データを読み込んで（ステップS600）、読み込んだデータに対して色変換処理を行う（ステップS602）。色変換処理を行うことにより、画像データはC、M、Y、K各色の階調値によって表現された階調データに変換される。

25 【0129】

ここで、前述した第1実施例のプリンタ200では、ドットの大きさを変えることはできず、各色毎にドットを形成するか形成しないかのいずれかの状態しか取り得ない。このため、色変換後のデータに基づいて直ちに、画素毎のドット形

成の有無を判断した。これに対して第3実施例のプリンタ200では、ドットの大きさを覚えて大ドット、中ドット、小ドットの3種類のドットを形成することが可能である。そこで、色変換処理によって得られた階調データを、一旦、大ドット用のデータ、中ドット用のデータ、小ドット用のデータに各色毎に変換する  
5 (ステップS604)。

#### 【0130】

階調データから、大ドット、中ドット、小ドットのそれぞれのドット密度データへの変換は、図21に示す変換テーブルを参照することによって行う。図示するように、変換テーブルには、大ドット密度データ、中ドット密度データ、小ド  
10 ット密度データが階調データに対応付けて記憶されており、かかる変換テーブルを参照することによって、色変換後の階調データを変換する。

#### 【0131】

次いで、大ドット密度データ、中ドット密度データ、小ドット密度データのそれぞれのデータに対して解像度変換処理を行う(ステップS606)。解像度変  
15 換には種々の方法を適用することができるが、説明を簡便にするために、ここでは第1実施例と同様に、画素を分割して解像度を変換するものとする。分割して生成された新たな画素には、元の画素の階調値と同じ階調値が設定される。

#### 【0132】

こうして解像度を印刷解像度に変換した後、個数データ生成処理を行う(ステップS608)。第3実施例ではプリンタ200が大ドット、中ドット、小ド  
20 ットの3種類のドットを形成することが可能である。このことに対応して個数データ生成処理では、これら各種ドットについてのドット個数のデータを生成して、得られたドット個数のデータをプリンタ200に出力する。

#### 【0133】

25 図22は、第3実施例の個数データ生成処理において、ドット密度データからドット個数のデータを生成する様子を示した説明図である。図22(a)は、画素群としてまとめられる各画素に、大中小の各種ドットについてのドット密度データが設定されている様子を模式的に表している。図中に示した実線の矩形の各



々が画素群を表している。画素群は複数の画素から構成されており、各画素にドット密度データが設定されているが、図示が煩雑となることを避けるために、図22では、個々の画素の表示は省略し、その代わりに画素群にドット密度データが設定されているものとして表示している。例えば、図22(a)の左上隅の画素群に、 $Data(L, M, S) = (2, 90, 32)$ と表示されているのは、この画素群の各画素に、大ドットのドット密度データ「2」、中ドットのドット密度データ「90」、小ドットのドット密度データ「32」が設定されていることを表している。もちろん、第1実施例においても説明したように、画素群を構成する全ての画素が同じ階調値を有する場合は、解像度変換処理を行わず、個数データ生成処理の中で実質的に解像度変換を行うこととしても良い。

#### 【0134】

第3実施例の個数データ生成処理では、こうした各種ドットについてのドット密度データに後述する処理を施すことにより、図22(b)に示すようなドット個数のデータを生成する。図22(b)においても図22(a)と同様に、実線の矩形は画素群を表している。画素群は複数の画素から構成されているが、各画素の表示は省略して、画素群毎にドット個数のデータが生成されているものとして表示している。例えば、図22(b)の左上隅の画素群中に、 $Dot(L, M, S) = (1, 2, 1)$ と表示されているのは、この画素群を構成する各画素については、大ドット、中ドット、小ドットのドット個数として、それぞれ「1」、「2」、「1」が生成されたことを表している。第3実施例の個数データ生成処理の内容については後述する。

#### 【0135】

プリンタ200は、コンピュータ100から出力されたドット個数のデータを受け取ると、画素位置決定処理を行う(ステップS610)。第3実施例では大ドット、中ドット、小ドットの3種類のドットを形成することから、画素位置決定処理では、これら各種ドットを形成する画素位置を決定する。第3実施例の画素位置決定処理の内容については後述する。

#### 【0136】

こうして画素位置を決定したら、プリンタ200はキャリッジ240の主走査と副走査とを繰り返しながらインク吐出用ヘッドを駆動することにより、印刷用紙上に大中小の各ドットを形成する（ステップS612）。この結果、画像データに対応した画像が印刷されることになる。

5           【0137】

D-2. 第3実施例の個数データ生成処理：

以下、第3実施例の個数データ生成処理の内容について説明する。図23は、第3実施例の個数データ生成処理の流れを示したフローチャートである。かかる処理も、コンピュータ100に内蔵されたCPUによって実行される。

10           【0138】

第3実施例の個数データ生成処理を開始すると、CPUは先ず初めに、画像データの中から所定の複数個の画素をまとめて画素群を生成する（ステップS700）。個数データ生成処理に先立って行われる解像度変換処理（図20のステップS606）において、第1実施例と同様に、画素を分割して新たな画素を生成  
15   することとしているから、ここでも、同じ画素から分割された複数の画素を画素群としてまとめるものとする。

            【0139】

次いで、画素群内の各画素について、大ドット、中ドット、小ドットのドット密度データを読み込む（ステップS702）。尚、ここでは、画素群を構成する  
20   画素は、いずれも同じ画素から分割されて同じ階調値を有するから、画素毎にドット密度データを読み込むのではなく、画素群毎に1画素分だけ読み込むこととしても良い。

            【0140】

こうして各種ドットのドット密度データを読み込んだら、ディザマトリックスを参照することにより、大ドット、中ドット、小ドットについての形成有無を判  
25   断する（ステップS704）。図24は、ディザマトリックスを参照しながら、着目画素に大ドット、中ドット、小ドットのいずれのドットを形成するべきかを判断する方法を示した説明図である。尚、図24では、ドット密度データとして

図2 2 (a) に示すデータを使用し、ディザマトリックスとしては、図8に示すマトリックスを使用する場合を想定している。

#### 【0141】

- 図2 4は、画像の左上隅にある画素群について、各種ドットの形成有無を判断する様子を示している。図中で、太い実線の矩形は画素群を表しており、画素群が細い破線で区分されているのは、画素群が複数の画素によって構成されていることを表している。また、画素の中に表された数値は、ディザマトリックスの対応する位置に設定されている閾値の値を表している。

#### 【0142】

- 10 ドット形成有無の判断を開始すると、先ず初めに、大ドットのドット密度データと、ディザマトリックスに設定されている閾値とを比較して、ドット密度データの方が大きい画素については大ドットを形成するものと判断する。図2 4 (a) は、こうして、画素群内の各画素について大ドットの形成有無を判断している様子  
15 示している。具体的には、大ドットのドット密度データは「2」であるから、一番左上隅にある階調値「1」の画素についてだけ、ドット密度データの方が大きく、他の画素については全てディザマトリックスの閾値の方が大きくなっている。そこで、この画素群では、大ドットは1つだけ形成される。図2 4 (a) 中  
20 で、ディザマトリックスの閾値が「1」の画素に細かい斜線が付されているのは、この画素に大ドットを形成すると判断されたことを表している。

#### 【0143】

- 大ドットについての形成有無を判断したら、今度は中ドットについての形成有無を判断する。中ドットについての判断に際しては、大ドットのドット密度データに中ドットのドット密度データを加算して、中ドット用の中間データを算出し、この中間データとディザマトリックスの閾値とを比較する。そして、中間データ  
25 の方が閾値よりも大きな画素については、中ドットを形成するものと判断する。このとき、既に大ドットを形成することとした画素については、中ドットの形成有無の判断は行わない。図2 4 (b) に即して具体的に説明すると、大ドットのドット密度データは「2」であり、中ドットのドット密度データは「90」であ

るから、中ドット用の中間データは「92」と算出される。この中間データとディザマトリックスの閾値とを比較する。但し、画素群中で一番左上隅にある画素については、既に大ドットを形成することとしているので、かかる比較は行わない。そして、ディザマトリックスの閾値が「42」の画素と「58」の画素について、中間データの方が大きくなるので、中ドットを形成すると判断する。  
図24(b)中で、これらの画素に斜線が付されているのは、中ドットを形成すると判断されたことを表している。

#### 【0144】

中ドットの形成有無を判断したら、最後に小ドットの形成有無を判断する。小ドットの判断に際しては、中ドット用の中間データに小ドットのドット密度データを加算して、小ドット用の中間データを算出し、未だドットを形成すると判断されていない画素について、この中間データとディザマトリックスの閾値とを比較する。そして、中間データの方が大きい画素については、小ドットを形成するものと判断する。図24(c)に即して具体的に説明すると、中ドット用の中間データは「92」であるから、小ドットのドット密度データ「32」を加算して、小ドット用の中間データは「124」と算出される。この中間データと、ディザマトリックスの閾値とを比較する。そして、ディザマトリックスの閾値が「109」の画素については中間データの方が大きいので、この画素には小ドットを形成すると判断する。図24(c)中で、この画素に粗い斜線が付されているのは、小ドットを形成すると判断されたことを表したものである。図23のステップS704では、以上のようにして画素群内の各画素について、大ドット、中ドット、小ドットの形成有無を判断する。

#### 【0145】

各種ドットの形成有無を判断したら、その画素群に形成すべき各種ドットの個数を記憶する(ステップS706)。図24に示した例では、大ドット、中ドット、小ドットの個数として、それぞれ1個、2個、1個である旨を記憶する。

#### 【0146】

こうして複数の画素を画素群にまとめて、画素群内に形成する各種ドットの個

数を記憶したら、画像データに含まれる全画素についての処理を終了したか否かを判断し（ステップS708）、未処理の画素が残っている場合は（ステップS708：no）、ステップS700に戻って新たな画素群を生成した後、続く一連の処理を繰り返す。こうして、全ての画素についての処理を終了したら（ステップS708：yes）、画素群毎に記憶しておいた各種ドットの個数をプリンタ200に向かって出力する（ステップS710）。ここで、プリンタ200へ出力するデータ量を更に少なくするために、各種ドットについての個数のデータは、次のようにしてコード化した状態で出力する。

#### 【0147】

10 図25は、各種ドットの個数のデータをコード化するために参照される変換テーブルを概念的に示した説明図である。変換テーブルには、大ドット、中ドット、小ドットの各種ドット個数の組合せ毎に、1つずつコード番号が設定されている。例えば、大ドット、中ドット、小ドットのいずれの個数も0個である組合せには、コード番号「0」が設定されている。また、大ドットと中ドットの個数が0個で  
15 あり、小ドットの個数が1個である組合せには、コード番号「1」が設定されている。

#### 【0148】

こうして、各種ドットの個数をコード番号に変換してからプリンタ200に出力することで、データ量を減らすことができる。この理由について若干説明する。  
20 ここでは画素群は8つの画素から構成されているものとしているから、各種ドットが取り得る最大個数は8個であり、従って、ドット個数のデータは、ドット種類毎に4ビットあれば足りる。ドットの種類は3種類だから、結局、画素群あたりに必要なデータ量は12ビットとなる。これに対して、図25に示すように、コード番号は0～165の値しか取り得ないから、コード化すれば画素群あたり  
25 8ビットあれば充分となる。すなわち、これだけでデータ量を2/3に減らすことが可能である。

#### 【0149】

更に、各種ドット個数の組合せの中には、実際の印刷では発生しない組合せも

多く含まれており、当然、これらの組合せにはコード番号を設定する必要はない。従って、必要な組合せにだけコード番号を設定することとすれば、画素群あたりに必要なデータ量は8ビットよりも少なくなるので、更にデータ量を減少させることが可能となる。

5           【0150】

図23のステップS710では、以上のような理由から、各種ドットについての個数のデータをコード化してプリンタ200に出力するのである。こうして、各種ドットの個数を画素群毎にコード化した状態でプリンタ200に出力したら、第3実施例の個数データ生成処理を終了して、図20の画像印刷処理に復帰  
10   する。

          【0151】

D-4. 第3実施例の変形例：

上述した第3実施例の個数データ生成処理にも種々の変形例が存在する。以下では、これら変形例について説明する。

15           【0152】

(1) 第1の変形例：

以上に説明した第3実施例の個数データ生成処理では、画素を特定しながら各種ドットの形成有無を判断したが、プリンタ200に供給される情報はドット個数の情報のみであり、画素位置の情報までは供給していない。このことから、前  
20   述した第1実施例の第1の変形例と同様に（図13参照）、画素位置を特定することなくドット個数を求めることとしても良い。図26は、こうして画素位置を特定することなく、各種ドットのドット個数を求める様子を概念的に示した説明図である。

          【0153】

25   図26は、図24での説明に用いた画素群を例にとって、画素位置を特定することなく各種ドットのドット個数を求める方法を示した説明図である。複数種類のドットについてドット個数を求める場合でも、単一のドットのドット個数を求める場合と同様に、簡易ディザマトリックスを使用する。前述したように簡易デ

ディザマトリックスとは、ディザマトリックス中では画素群内の各画素に対応付けられている閾値が、各画素と対応付けられることなく画素群毎にひとまとまりの閾値の組として記憶されているマトリックスである。例えば図24に示した画素群には、画素群内の各画素に対応付けてディザマトリックスの閾値が設定されている。これに対して簡易ディザマトリックスを用いる図26では、これら閾値の組{1, 42, 58, 109, 170, 177, 212, 255}が画素群に設定されている。以下、図26に従って、画素位置を特定することなく各種ドットのドット個数を求める方法について説明する。

#### 【0154】

- 10     ドット個数は、先ず初めに大ドットのドット個数を求め、次に中ドット、最後に小ドットの順番で求めていく。図26(a)は大ドットのドット個数を求める様子を表しており、図26(b)は中ドットのドット個数を、図26(c)は小ドットのドット個数を求める様子を表している。大ドットのドット個数を求めるためには、大ドットのドット密度データと、画素群に設定されている閾値の組とを比較する。そして、ドット密度データより小さな閾値の個数だけ大ドットを形成すると判断する。図26(a)に示した例では、大ドットのドット密度データは「2」であるから、ドット密度データより小さな閾値は1つだけ存在しており、従って、大ドットのドット個数を1個と求めることができる。図26(a)中で閾値「1」に細かい斜線が施されているのは、この閾値については大ドットを形成すると判断されたことを模式的に表したものである。
- 15     20

#### 【0155】

- 大ドットのドット個数を求めたら、大ドットのドット密度データに中ドットのドット密度データを加えて中ドット用の中間データを算出する。そして、この中間データと閾値の組とを比較して、中間データよりも小さな閾値の個数だけ中ドットを形成すると判断する。但し、既に大ドットを形成すると判断されている閾値との比較は行わない。図26(b)に示した例では、大ドットのドット密度データは「2」であり、中ドットのドット密度データは「90」であるから、中ドット用の中間データは「92」と算出される。既に大ドットを形成すると判断し
- 25

た閾値「1」を除けば、この中間データ「92」よりも小さな閾値は2つ存在する。そこで、この画素群に形成する中ドットの個数は2個であると求めることができる。図26(b)中に、閾値「42」と閾値「58」に斜線が施されているのは、これら閾値については中ドットを形成すると判断されていることを模式的に表したものである。

#### 【0156】

中ドットのドット個数を求めたら、中ドット用の中間データに小ドットのドット密度データを加えて、小ドット用の中間データを算出する。そして、この中間データと閾値の組とを比較することにより、小ドットのドット個数を求めてやる。

10 図26(c)に示した例では、中ドット用の中間データは「92」、小ドットのドット密度データは「32」であるから、小ドット用の中間データは「124」となる。いずれのドットも形成されていない閾値の中で、この中間データよりも小さな閾値は1つだけあるから、この画素群に形成される小ドットの個数は1個だけであると判断することができる。図26(c)中で、閾値「109」に斜線

15 が施されているのは、この閾値については小ドットを形成すると判断したことを模式的に表したものである。

#### 【0157】

このように、簡易ディザマトリックスを参照しながら、各種ドットのドット個数を求めて行けば、画素位置を特定することなくドット個数を求めることが可能

20 である。こうすれば、ドット密度データあるいは中間データを、閾値の組と比較するだけでドット個数を求めることができ、画素毎に設定された閾値と比較する必要がないので、各種ドットについてのドット個数のデータを迅速に求めることが可能となる。

#### 【0158】

25 (2) 第2の変形例：

また、上述した第1の変形例のように、複数の閾値が、閾値組の中での大きさの順番を示す情報とともに記憶されている場合は、大ドットのドット密度データ、あるいは中ドット用の中間データなどを、次のようにして閾値と比較することに



より、各種ドットの個数をより迅速に決定することも可能である。以下では、こうした第2の変形例について説明する。

#### 【0159】

図27は、第3実施例の第2の変形例における個数データ生成処理の流れを示すフローチャートである。第2の変形例の個数データ生成処理を開始すると、先ず初めに、所定の複数個の画素をまとめて画素群を生成し（ステップS750）、画素群内の各画素についての大ドット、中ドット、小ドットのドット密度データを読み込む（ステップS752）。尚、前述した第3実施例と同様に、ここでも、画素群を構成する各画素は、いずれも同じドット密度データを有しているから、  
10 画素毎にデータを読み込むのではなく画素群毎に1画素分だけドット密度データを読み込むこととしても良い。

#### 【0160】

こうして、大ドット、中ドット、小ドットのドット密度データを読み込んだら、先ず初めに画素群内に形成する大ドットの個数を決定し（ステップS754）、  
15 次いで中ドットの個数を決定した後（ステップS756）、最後に小ドットの個数を決定する処理を行う（ステップS758）。詳細には後述するが、第2の変形例においては、これら各種ドットの個数を決定するに際して、後述する方法を採用しているために、ドットの個数を迅速に決定することが可能となっている。

#### 【0161】

20 大中小ドットの個数を決定したら、その画素群に形成すべき各種ドットの個数を記憶し（ステップS760）、画像データに含まれる全画素についての処理を終了したか否かを判断する（ステップS762）。そして、未処理の画素が残っている場合は（ステップS762：no）、ステップS750に戻って新たな画素群を生成した後、続く一連の処理を繰り返す。こうして、全ての画素について  
25 の処理を終了したら（ステップS762：yes）、画素群毎に記憶しておいた各種ドットの個数をコード化してプリンタ200に出力する（ステップS764）。各種ドットの個数をコード化する方法については、前述した第3実施例と同様であり、ここでは説明を省略する。

### 【0162】

次に、大ドット、中ドット、小ドットの個数を決定する処理について、順番に説明する。図28は、大ドットの個数を決定する処理の流れを示したフローチャートである。第3実施例の第2の変形例では、大ドットの個数を決定する処理を  
5 開始すると先ず初めに、直前に決定した大ドットの個数 DotL、すなわち既にドット個数を決定した画素群の中から直前に決定した画素群についての大ドットの個数を取得する（ステップS770）。

### 【0163】

次いで、処理している画素群の閾値組の中で DotL 番目の閾値（小さい方から  
10 数えて DotL 番目の閾値）と、大ドットのドット密度データとを比較する（ステップS772）。そして、閾値の方が小さい場合は、次の順番の閾値、すなわち DotL+1 番目の閾値と、大ドットのドット密度データとを比較する（ステップS774）。図26に示したように、閾値組には複数の閾値が小さな閾値から順番に並んだ状態で記憶されており、DotL+1 番目の閾値は DotL 番目の閾値よりも大  
15 きな閾値となっている。次の順番の閾値についても、依然として閾値の方が小さいと判断された場合は、DotL の値を1つインクリメントした後（ステップS776）、ステップS774に戻って、更に次の順番の閾値、すなわち DotL+2 番目の閾値と大ドットのドット密度データとを比較する。こうして大ドットのドット密度データを次第に大きな閾値と比較していき、ついには閾値の方が大きくな  
20 ったら、そのときの DotL の値を大ドットの形成個数と決定する（ステップS778）。

### 【0164】

一方、ステップS772において、大ドットのドット密度データよりも DotL 番目の閾値の方が大きいと判断された場合は、1つ前の順番の閾値、すなわち  
25 DotL-1 番目の閾値と、大ドットのドット密度データとを比較する（ステップS780）。順番が1つ前になると、閾値の大きさは少しだけ小さくなる。順番が1つ前の閾値と比較しても、依然として閾値の方が大きいと判断された場合は、DotL の値を1つデクリメントした後（ステップS782）、ステップS780に

- 戻って、更に1つ前の閾値、すなわち DotL-2 番目の閾値と大ドットのドット密度データとを比較する。こうして大ドットのドット密度データを次第に小さな閾値と比較していき、ついには閾値の方が小さくなったら、そのときの DotL の値を大ドットの形成個数と決定する（ステップS 7 7 8）。以上のようにして、大
- 5 ドットの個数を決定したら、図 2 8 の大ドット個数決定処理を終了して、図 2 7 に示した第 2 の変形例の個数データ生成処理に復帰する。

#### 【0 1 6 5】

- このように第 2 の変形例においても前述した第 1 の変形例と同様に、閾値組に含まれる複数の閾値の中で、大ドットのドット密度データより小さな閾値の個数
- 10 を検出し、この個数を大ドットの個数と決定する。ここで第 2 の変形例では、ドット密度データを閾値と比較するに際して、初めに比較する閾値を、直前に処理した画素群の大ドットの個数 DotL に基づいて決定する。例えば、上述した説明では、直前に処理した画素群の大ドットの個数が DotL 個である場合は、閾値の順番が DotL 番目の閾値から比較を始める。こうすれば、次の理由から僅かな回
- 15 数比較するだけで、大ドットの個数を決定することが可能である。

#### 【0 1 6 6】

- 一般に、画像データは、隣接する画素間では近似する値を取る場合が多く、従って、隣接する画素群では近似したドット個数となる傾向がある。すなわち、ある画素群で大ドットの形成個数が DotL 個と判断された場合は、次の画素群でも
- 20 大ドットの形成個数は DotL 個前後の個数と判断される可能性が高い。このことから、初めに DotL 番目の閾値から比較を開始して、閾値の方が大きい場合は次第に小さな閾値と比較し、逆に閾値の方が小さい場合は次第に大きな閾値と比較していけば、大ドットのドット密度データより小さな閾値の個数を、僅かな比較回数で決定することが可能となる。尚、ここでは、直前の画素群に形成される大
- 25 ドットの個数が DotL 個である場合に、DotL 番目の閾値から比較を開始するものとして説明した。しかし、比較を開始する閾値の番数をドットの個数に基づいて決定するのであれば、ドットの個数と閾値の番数とは一致している必要はなく、例えば、ドット個数が DotL 個の場合に DotL+1 番目の閾値から比較を開始した

り、あるいは DotL-1 番目の閾値から比較を開始することとしても構わない。

#### 【0167】

以上のようにして大ドットの個数を決定したら、続いて中ドットの個数を決定する処理を開始する（図27のステップS756）。図29は、第2の変形例に  
5 おいて中ドットの個数を決定する処理の流れを示したフローチャートである。以下、フローチャートに従って中ドットの個数を決定する処理について説明する。

#### 【0168】

中ドットの個数を決定する処理を開始すると、先ず初めに中ドットを決定するための中間データを算出する（ステップS780）。中ドットを決定するための  
10 中間データは、前述した第3実施例と同様に、大ドットのドット密度データに中ドットのドット密度データを加算することによって算出することができる。

#### 【0169】

次いで、その画素群について決定した大ドットの個数 DotL が0個か否かを判断する（ステップS782）。決定した大ドットの個数 DotL が0個である場合は  
15 は（ステップS782：yes）、直前に処理した画素群の中ドットの個数 DotM を取得する（ステップS784）。決定した大ドットの個数 DotL が0個ではない場合は（ステップS782：no）、大ドットの個数 DotL を1つだけインクリメントして、この値を仮に中ドットの個数 DotM とする（ステップS786）。

#### 【0170】

20 こうして、仮の中ドットの個数 DotM を設定した後、処理している画素群の閾値組の中で DotM 番目の閾値（すなわち、小さい方から数えて DotM 番目の閾値）と、先に算出した中間データとを比較する（ステップS788）。そして、閾値の方が小さい場合は、次の順番の閾値、すなわち DotM+1 番目の閾値と、中間データとを比較する（ステップS790）。次の順番の閾値についても、依然とし  
25 て閾値の方が小さいと判断された場合は、DotM の値を1つインクリメントした後（ステップS792）、ステップS790に戻って、更に次の順番の閾値、すなわち DotM+2 番目の閾値と中間データとを比較する。こうして中ドット用の中間データを次第に大きな閾値と比較していき、ついには閾値の方が大きくなった

ら、そのときの DotM の値から、その画素群に形成される大ドットの個数 DotL を引いた値 ( $\text{DotM} - \text{DotL}$ ) を中ドットの形成個数と決定する (ステップ S 7 9 4)。

#### 【0171】

- 5 一方、ステップ S 7 8 8 において、中ドット用の中間データよりも DotM 番目の閾値の方が大きいと判断された場合は、1 つ前の順番の閾値、すなわち DotM-1 番目の閾値と、中間データとを比較する (ステップ S 7 9 6)。順番が1つ前の閾値と比較しても、依然として閾値の方が大きいと判断された場合は、DotM の値を1つデクリメントした後 (ステップ S 7 9 8)、ステップ S 7 9 6 に戻って、
- 10 更に1つ前の閾値、すなわち DotM-2 番目の閾値と中間データとを比較する。こうして中ドット用の中間データを次第に小さな閾値と比較していき、ついには閾値の方が小さくなったら、そのときの DotM の値から、その画素群に形成される大ドットの個数 DotL を引いた値 ( $\text{DotM} - \text{DotL}$ ) を中ドットの形成個数と決定する (ステップ S 7 9 4)。以上のようにして、中ドットの個数を決定したら、
- 15 図 2 9 の中ドット個数決定処理を終了して、図 2 7 に示した第 2 の変形例の個数データ生成処理に復帰する。

#### 【0172】

- このように第 2 の変形例においても前述した第 1 の変形例と同様に、閾値組に含まれる複数の閾値の中で、中ドット用の中間データより小さな閾値の個数を検
- 20 出し、この個数に基づいて中ドットの個数と決定する。中間データと閾値とを比較するに際しては、その画素群に大ドットを形成しない場合は、直前に処理した画素群の中ドットの個数に基づいて決定した順番の閾値から比較を開始する。前述したように、隣接する画素群に形成される各種ドットの個数は近似した個数となる傾向があるから、その画素群に大ドットが形成されない場合は、直前にドッ
- 25 トの個数を決定した画素群でも大ドットが形成されていない確率が高く、そして、その画素群に形成される中ドットの個数は、直前に決定した中ドットの個数と近似した個数となる確率が高いと考えられる。そこで、中間データと最初に比較する閾値を、直前に決定した中ドットの個数に基づいて決定してやれば、僅かな比

較回数で中ドットの個数を決定することが可能となる。

#### 【0173】

また、その画素群に大ドットを形成すると判断されている場合は、大ドットの  
個数 DotL よりも1つだけ番号が大きい閾値、すなわち DotL+1 番目の閾値から  
5 比較を開始する。これは次の理由によるものである。中ドットの個数を決定する  
ための中間データは、大ドットのドット密度データと中ドットのドット密度デー  
タとを加算したデータであり、大ドットのドット密度データよりも大きなデータ  
である。従って、大ドットの個数が DotL 個と決定されている場合は、閾値組の  
中で DotL 番目の閾値は比べるまでも無く中間データより小さいことは明らかで  
10 ある。そこで、直ちに DotL+1 番目の閾値から比較を開始してやれば、中間デー  
タより小さな閾値の個数を迅速に決定することができ、延いては中ドットの個数  
DotM を迅速に決定することが可能となるのである。

#### 【0174】

以上のようにして中ドットの個数を決定したら、続いて小ドットの個数を決定  
15 する処理を開始する（図27のステップS758）。図30は、第2の変形例に  
おいて小ドットの個数を決定する処理の流れを示したフローチャートである。図  
30に示した小ドットの個数を決定する処理は、前述した図29の中ドットの個  
数を決定する処理に比べて、その画素に他のドットが形成されているか否かに関  
わらず、同じ処理を行う点が大きく異なっている。これは、大ドットが形成され  
20 ない確率に比べて、大ドットも中ドットもいずれも形成されない確率が小さいた  
めに、これらドットがいずれも形成されない場合に特別な処理を行う利点が比較  
的少ないことによるものである。もちろん、前述した中ドットの場合と同様に、  
大ドットも中ドットも形成されない場合には、直前に決定した小ドットの個数に  
基づいて、初めに比較する閾値を決定することとしても構わない。

#### 25 【0175】

小ドットの個数を決定する処理を開始すると、先ず初めに小ドットを決定する  
ための中間データを算出する（ステップS800）。小ドットを決定するための  
中間データは、中ドット用の中間データに小ドットのドット密度データを加算す

ることによって算出することができる。

#### 【0176】

次いで、その画素群について既に決定した大ドットの個数  $DotL$  と中ドットの個数  $DotM$  と加算値を1つだけインクリメントして、この値を、仮の小ドットの  
5 個数  $DotS$  とする（ステップS802）。

#### 【0177】

次いで、処理している画素群の閾値組の中で  $DotS$  番目の閾値（すなわち、小さい方から数えて  $DotS$  番目の閾値）と、先に算出した中間データとを比較する（ステップS804）。閾値の方が小さい場合は、次の順番の閾値すなわち  $DotS+1$   
10 番目の閾値と中間データとを比較し（ステップS806）、この閾値についても、依然として閾値の方が小さいと判断された場合は、 $DotS$  の値を1つインクリメントした後（ステップS808）、ステップS806に戻って、更に次の順番の閾値、すなわち  $DotS+2$  番目の閾値と中間データとを比較する。こうして小ドット用の中間データを次第に大きな閾値と比較していき、ついには閾値の方が大き  
15 くなったら、 $DotS - DotM - DotL$  を算出して、この値を小ドットの形成個数と決定する（ステップS810）。

#### 【0178】

一方、ステップS804において、小ドット用の中間データよりも  $DotS$  番目の閾値の方が大きいと判断された場合は、1つ前の順番の閾値、すなわち  $DotS-1$   
20 番目の閾値と、中間データとを比較する（ステップS812）。順番が1つ前の閾値と比較しても、依然として閾値の方が大きいと判断された場合は、 $DotS$  の値を1つデクリメントした後（ステップS814）、ステップS812に戻って、更に1つ前の閾値、すなわち  $DotS-2$  番目の閾値と中間データとを比較する。こうして小ドット用の中間データを次第に小さな閾値と比較していき、ついには閾  
25 値の方が小さくなったら、そのときの  $DotS - DotM - DotL$  を算出して、この値を小ドットの形成個数と決定する（ステップS810）。以上のようにして、小ドットの個数を決定したら、図30の小ドット個数決定処理を終了して、図27に示した第2の変形例の個数データ生成処理に復帰する。

### 【0179】

このように小ドット用の中間データと閾値とを比較するに際しては、中ドット用の中間データと比較した最後の閾値、すなわち、大きさの順番が  $\text{DotL} + \text{DotM}$  番目の次の閾値から比較を開始する。前述したように小ドット用の中間データは、  
5 中ドット用の中間データに小ドットのドット密度データを加算したデータであるから、閾値組の中で大きさの順番が  $\text{DotL} + \text{DotM}$  番目の閾値は、比べるまでもなく、小ドット用の中間データよりも小さいことが明らかである。そこで、直ちに  $\text{DotL} + \text{DotM} + 1$  番目の閾値から比較を開始することで、小ドットの個数  $\text{DotM}$  を迅速に決定することが可能となる。

### 10 【0180】

D-3. 第3実施例の画素位置決定処理：

以上に説明したように、第3実施例の個数データ生成処理では、画素群毎に各種ドットのドット個数が求められた後、ドット個数を示すデータがプリンタ200に転送される。プリンタ200では、転送されてきたデータを受け取って、各種ドットを形成する画素位置を決定した後、決定に従って印刷用紙上に各種ドットを形成することによって画像を印刷する。以下では、ドット個数を示すデータ  
15 に基づいて各種ドットを形成する画素位置を決定する処理について説明する。

### 【0181】

図31は、第3実施例の画素位置決定処理の流れを示したフローチャートである。かかる処理は、プリンタ200の制御回路260に内蔵されたCPUによって実行される。  
20

### 【0182】

第3実施例の画素位置決定処理を開始すると、まず初めに、画素位置を決定しようとする画素群を選択し（ステップS850）、次いで、転送されてきたデータの中から、選択した画素群のデータを取得する（ステップS852）。ここでは、図25を用いて説明したように、ドット個数のデータはコード化された状態で転送されているものとする。そこで、コード化されたデータを、各種ドットのドット個数を示すデータに復号化する（ステップS854）。  
25



### 【0183】

データの復号化は、図25に示した変換テーブルを、コード番号の側から参照することによって行う。例えば、コード化されたデータが「163」であれば、変換テーブル中でコード番号「163」に対応付けられているドット個数から、  
5 大ドット7個、中ドット0個、小ドット1個とドット個数のデータに復号することができる。

### 【0184】

こうして各種ドットのドット個数が得られたら、ディザマトリックスを参照することにより、各種ドットを形成する画素位置を決定する（ステップS856）。  
10 かかる処理について、図32を参照しながら説明する。

### 【0185】

図32は、ある画素群について、各種ドットのドット個数が与えられたときに、ディザマトリックスを参照しながら、これらドットを形成する画素位置を決定する様子を示した説明図である。図中に示した太い実線の矩形は画素群を表している。  
15 画素群を区分する細い破線は、画素群が複数の画素から構成されていることを表している。また、画素中に示された数値は、ディザマトリックスの対応する位置に設定されている閾値を示している。尚、ディザマトリックスは、ドット個数を求めるために用いたマトリックスと同じマトリックスを使用するものとする。

### 20 【0186】

今、この画素群に形成する各種ドットの個数が、大ドット1個、中ドット2個、小ドット1個であるものとして、初めに大ドットを形成する画素位置を決定する。前述したように、ディザマトリックスの閾値は、ドットの形成され易さを表していると考えることができるから、大ドットを1個だけ形成するとすれば、最も小  
25 さい閾値の設定された画素に形成されることになる。大ドットの画素位置を決定したら、続いて中ドットを形成する画素位置を決定する。中ドットは2個形成することになっており、また、最も小さい閾値の画素には既に大ドットが形成されているから、閾値が2番目に小さい画素と3番目に小さい画素の2つの画素に中

ドットを形成すると判断する。中ドットに続いて小ドットの画素位置を決定する。小ドットは1個だけ形成することになっており、また、閾値の最も小さな画素から3番目に小さな画素までは、既に大ドットあるいは中ドットが形成されているから、4番目に閾値の小さな画素に小ドットを形成すると判断する。

5       【0187】

図32は、こうして大ドット、中ドット、小ドットの順番で、ドットを形成する画素を決定している様子を表しており、図中で、細かい斜線が付された画素は大ドットを形成すると判断された画素を、中間の斜線が付された画素は中ドットを形成すると判断された画素を、粗い斜線が付された画素は小ドットを形成すると判断された画素を、それぞれ表している。図31のステップS856では、このようにして、ディザマトリックスを参照しながら、各種ドットを形成する画素位置を決定する。

      【0188】

尚、以上の説明では、コード化されたデータを復号化する際に、各種ドットの  
15   ドット個数のデータに復号化するものとして説明した。しかし、上述したように、大ドット、中ドット、小ドットの順番で画素位置を決定することから、各種ドットのドット個数ではなく、大ドットの個数、大ドットと中ドットの合計個数、大ドットと中ドットと小ドットの合計個数に復号化することとしても良い。例えば、図32に示した例では、{大ドット：1個、中ドット：2個、小ドット：1個}  
20   と復号化する代わりに、{大ドット：1個、大ドット+中ドット：3個、大ドット+中ドット+小ドット：4個}と復号化するのである。

      【0189】

図33は、このような復号化を行うために参照される復号化テーブルを概念的に表した説明図である。このように復号化しておけば、次のように画素位置を決定する処理を迅速化することができる。例えば、中ドットの画素位置を決定する場合について説明すると、大ドット+中ドットの個数は3個と復号化されているので、設定されている閾値が小さい方から3番目までの画素を選択する。そして、既に他のドット（大ドット）が形成されている画素を除いて、選択した画素に中

ドットを形成すると判断することができる。こうすれば、既に他のドットを形成すると判断した画素数を考慮することなく、画素位置を決定することができるので、処理の迅速化を図ることが可能となる。

#### 【0190】

- 5 以上のようにして、1つの画素群について、各種ドットを形成する画素位置を決定したら（図31のステップS856）、コンピュータ100から供給された全画素群のデータについて、画素位置を決定する処理を終了したか否かを判断する（ステップS858）。未処理の画素群が残っている場合は（ステップS858：no）、ステップS850に戻って、新たな画素群について続く一連の処理
- 10 を繰り返す。こうして全ての画素群について画素位置を決定したと判断されたら（ステップS858：yes）、図31に示す画素位置決定処理を抜けて、画像印刷処理に復帰した後、印刷用紙上に各種ドットを形成する。この結果、画像データに対応した画像が印刷されることになる。

#### 【0191】

- 15 プリンタ200がいわゆる多値プリンタである場合には、以上に説明した第3実施例の画像印刷処理を行うことにより、コンピュータ100からプリンタ200に各種ドットのドット個数を転送しながら画像を印刷することができる。こうすれば、画素毎にドット形成の有無を示すデータを供給するよりも、迅速に供給することができるので、たとえ、画素数の多い画像でも、迅速に画像を印刷する
- 20 ことが可能となる。

#### 【0192】

E. その他の変形例：

本発明には、上述した実施例の他にも、更に各種の変形例が存在する。以下では、これら変形例について簡単に説明する。

- 25 【0193】

(1) 第1の変形例：

上述した各種の実施例の個数データ生成処理では、いわゆるディザ法を用いてドット形成の有無を判断している。しかし、画像データの階調値に応じて適切な

密度となるようにドット個数を求めることができるのであれば、どのような手法を使用することもできる。

#### 【0194】

例えば、図34に示した変形例の個数データ生成処理のように、画素群内の各  
5 画素の平均階調値を算出して、この平均階調値に基づいて、画素群内に形成する  
ドット個数を一義的に求めることとしても良い。以下、図34のフローチャート  
に従って簡単に説明する。

#### 【0195】

個数データ生成処理を開始すると、先ず初めに、所定の複数個の画素を画素群  
10 としてまとめて（ステップS900）、画素群内の各画素の平均階調値を算出し  
（ステップS902）、この平均階調値に基づいて画素群内に形成するドット個  
数を一義的に決定する。すなわち、図35に例示したように、画素群内に形成す  
るドット個数を平均階調値に対応付けて予め定めておき、この対応関係に基づい  
て、画素群の平均階調値からドット個数を決定する。尚、図35では、ドット種  
15 類が3種類である場合について示しているが、ドット種類は3種類に限られるも  
のではない。

#### 【0196】

次いで、こうして決定したドット個数を画素群毎に記憶した後（ステップS9  
06）、全画素についての処理が終了したか否かを判断する（ステップS908）。  
20 そして、未処理の画素が残っていれば、ステップS900に戻って続く一連の処  
理を繰り返す。全ての画素の処理を終了したと判断されたら、記憶しておいたド  
ット個数の図33を用いて説明した方法によってコード化した後、コード化した  
ドット個数のデータをプリンタ200に向かって出力する（ステップS910）。

#### 【0197】

25 プリンタ200では、コンピュータ100から出力されたドット個数のデータ  
を受け取った後、前述した画素位置決定処理を行って画像を印刷する。こうすれ  
ば、画素群毎のドット個数のデータを簡便に求めることができるので、ドット個  
数のデータを迅速に出力することができ、延いては、更に迅速に画像を印刷する

ことが可能となる。

【0198】

(2) 第2の変形例：

以上に説明した各種実施例の個数データ生成処理では、初めに、複数の画素  
5 を画素群としてまとめた後、画素群内に形成するドット個数を決定している。しかし、初めに、画素毎のドット形成の有無を判断し、その後に、複数の画素を画素群としてまとめることとしても良い。

【0199】

例えば、図36に示した変形例では、初めに、画像データにいわゆる誤差拡散  
10 法を適用して、全ての画素についてドット形成の有無を判断する（ステップS950）。その後、所定の複数の画素を画素群としてまとめ（ステップS952）、形成するドット個数を画素群毎に数えて記憶する（ステップS954）。

【0200】

こうして1つの画素群についてドット個数を記憶したら、全ての画素について  
15 の処理を終了したか否かを判断し（ステップS956）、未処理の画素が残っている場合は、ステップS900に戻って続く一連の処理を繰り返す。全ての画素についての処理を終了したと判断されたら、画素群毎に記憶しておいたドット個数をプリンタ200に向かって出力する（ステップS958）。

【0201】

20 プリンタ200では、こうしてコンピュータ100からドット個数のデータを受け取ると、前述した画素位置決定処理を行って、ドットを形成する画素位置を画素群毎に決定する。尚、図36に示す変形例では誤差拡散法に基づいてドット個数を求めているが、このような場合でも、前述したようにディザマトリックスを参照しながら画素位置を決定することができる。

25 【0202】

また、一般に、誤差拡散法を用いてドット形成の有無を判断した場合、ドットの形成密度が低い領域でノイズの少ない良好な画質が得られることが知られている。このことから、例えば、ドットが1つも形成されない多くの画素群の中に、

ドットの形成される画素群がまばらに分布しているようなドット密度の低い画像を印刷する場合、画素群のドット個数を誤差拡散法に基づいて求めてやれば、ドットを適切に分布させて、高画質な画像を得ることが可能となる。

#### 【0203】

##### 5 (3) 第3の変形例:

以上に説明した各種実施例では、ディザマトリックスに基づいて画素位置を決定するものとして説明した。前述した序列マトリックスを参照しながら画素位置を決定する場合は、直接にはディザマトリックスを参照しているわけではないが、序列マトリックスはディザマトリックスに基づいて生成されていることから、間  
10 接的には、ディザマトリックスに基づいて画素位置を決定している考えることができる。しかし、画素群毎に、異なる序列を用いて画素位置を決定することができれば、必ずしもディザマトリックスに基づいて決定する必要はない。

#### 【0204】

例えば、図37に示したように、複数の序列を記憶しておき、この中から画素  
15 毎に適当な序列を選択して画素位置を決定することとしても良い。以下、図37のフローチャートに従って簡単に説明する。

#### 【0205】

プリンタ200の制御回路260に内蔵されたCPUは、変形例の画素位置決定処理を開始すると先ず初めに、画素位置を決定しようとする画素群を選択して、  
20 この画素群のドット個数のデータを取得する(ステップS970)。次いで、予め記憶しておいた複数の序列の中から、任意の序列を1つ選択する(ステップS972)。制御回路260のROMには、画素群内の各画素について複数通りの序列が予め記憶されている。図38は、ROMに記憶されている複数通りの序列を模式的に示した説明図である。ステップS972では、これら序列の中から序  
25 列を1つ選択する。

#### 【0206】

そして、選択した序列を参照しながら、画素群内にドットを形成する画素位置を決定する(ステップS974)。こうして1つの画素群について画素位置を決

定したら、全ての画素群についての処理を終了したか否かを判断し（ステップS 976）、未処理の画素群が残っていれば、ステップS 970に戻って続く一連の処理を行って、画素位置を決定する。そして、全ての画素群についての処理を終了するまで、上述した処理を繰り返す。

5           【0207】

この様にしても、画素群毎のドット個数のデータから、ドットを形成する画素位置を決定することができる。また、画素位置を決定するために参照する序列は、ほとんどの場合、画素群毎に異なる序列が選択されるので、同じパターンでドットが形成されて画質が悪化するおそれもない。

10           【0208】

以上、各種の実施例について説明してきたが、本発明は上記すべての実施例に限られるものではなく、その要旨を逸脱しない範囲において種々の態様で実施することができる。例えば、上述の機能を実現するソフトウェアプログラム（アプリケーションプログラム）を、通信回線を介してコンピュータシステムのメイン  
15   メモリまたは外部記憶装置に供給し実行するものであってもよい。もちろん、C D-R O Mやフレキシブルディスクに記憶されたソフトウェアプログラムを読み込んで実行するものであっても構わない。

            【0209】

また、以上の実施例では、印刷用紙上にドットを形成して画像を印刷するプリンタに本発明を適用した場合について説明したが、本発明の適用範囲はプリンタ  
20   に限られるものではなく、例えば、液晶表示画面上で輝点を適切な密度で分散させることにより、階調が連続的に変化する画像を表現する液晶表示装置などにも、本発明を好適に適用することができる。